AD-A217 805

DTIC ACCESSION NUMBER

PHOTOGRAPH THIS SHEET

DTIC FILE COPY

LEVEL

INVENTORY: 1

GL-TR-89-0137

DOCUMENT IDENTIFICATION

May 1989

DISTRIBUTION STATEMENT A

Approved for public release; Distribution Unlimited

DISTRIBUTION STATEMENT

ACCESSION FOR		
NTIS	GRA&I	☒
DTIC	TAB	☐
UNANNOUNCED		☐
JUSTIFICATION		
BY		
DISTRIBUTION /		
AVAILABILITY CODES		
DIST	AVAIL AND/OR SPECIAL	
A-1		

DISTRIBUTION STAMP

DTIC COPY INSPECTED 6

DTIC
ELECTE
FEB 09 1990
S E D

DATE ACCESSIONED

DATE RETURNED

90 02 06 274

DATE RECEIVED IN DTIC

REGISTERED OR CERTIFIED NO.

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-FDAC

DTIC FORM 70A
MAR 86

DOCUMENT PROCESSING SHEET

PREVIOUS EDITION MAY BE USED UNTIL STOCK IS EXHAUSTED.

AD-A217 805

GL-TR-89-0137

UV Spectrometer System AFGL 801 A HUP

J. C. E. Berends
R. C. Eppig
J. T. Riley

Research Support Instruments, Inc
10610 Beaver Dam Road
Cockeysville, MD 21030

May 1989

Final Report
12 January 1979-31 October 1988

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

GEOPHYSICS LABORATORY
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
HANSCOM AIR FORCE BASE, MASSACHUSETTS 01731-5000

"This technical report has been reviewed and is approved for publication"

Charles A. Forsberg

CHARLES A. FORSBERG
Contract Manager

Robert E. Huffman

ROBERT E. HUFFMAN
Branch Chief

FOR THE COMMANDER

ROBERT A. SKRIVANEK
Division Director

This report has been reviewed by the ESD Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS).

Qualified requestors may obtain additional copies from the Defense Technical Information Center. All others should apply to the National Technical Information Service.

If your address has changed, or if you wish to be removed from the mailing list, or if the addressee is no longer employed by your organization, please notify AFGL/DAA, Hanscom AFB, MA 01731. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it be returned.

REPORT DOCUMENTATION PAGE

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	1989 May	Final Report(12 Jan 1979-31 Oct 1988)

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
UV Spectrometer System AFGL 801 A HUP	62101F 669017AA

6. AUTHOR(S)	
J. C. E. Berends R. C. Eppig J. T. Riley	Contract F19628-79-C-0044

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Research Support Instruments, Inc 10610 Beaver Dam Road Cockeysville, MD 21030	

9. SPONSORING MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING MONITORING AGENCY REPORT NUMBER
Geophysics Laboratory Hanscom AFB Massachusetts 01731-5000 Contract Manager: Charles A. Forsberg/LIM	GL-TR-89-0137

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT

Approved for public release;
Distribution unlimited

13. ABSTRACT (Maximum 200 words)

Seven (7) spectrometers, covering the ultraviolet and visible spectral regions were designed, built, and tested under this contract. One (1) UV Spectrometer was packaged with a telescope, scan platform and electronics, and flown on Shuttle Mission STS-4. Several other spectrometers were similarly packaged for a subsequent Shuttle experiment. Ground support equipment (GSE) was provided for the units.

14. SUBJECT TERMS			15. NUMBER OF PAGES
	Space Shuttle	Horizon	
Spectrometers	Flight Electronics	Telescope	378
Ultraviolet	Ground Support Equipment(GSE)	Ebert-Fastie	16. PRICE CODE
Instrumentation	HUP	Scan Platform	

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Same as Report

TABLE OF CONTENTS

<u>LIST OF APPENDICIES</u>

LIST OF ILLUSTRATIONS

1.0 INTRODUCTION

This contract called for the design, fabrication, testing and launch preparation of an ultraviolet spectrometer system for use on spacecraft. The major components of the system are the spectrometers, the detectors, the telescope, the scanning mechanism and the ground support equipment. A photograph of the spectrometer with the telescope attached is shown in FIGURE 1. The contract initially called for a prototype to be constructed followed by six (6) flight units co-aligned on a scanning mechanism. At the time the contract was let, the spacecraft had not been determined.

In March 1981 it was determined that the prototype would be used on a shuttle orbiter flight. After the successful flight of the prototype unit, the contract was changed so that a package of two (2) units would be prepared for another shuttle orbiter flight. In addition, the ground support equipment would be modified to support the two unit package during the flight. Upon completion of this mission, the two units would be refurbished and mounted in the package of six spectrometer/telescope units as originally proposed.

In September 1984, due to the unavailability of the detector which was used on the prototype, it was necessary to redesign the spectrometer. Because of the difference in physical size between the two detectors, an extensive redesign of the instruments was required. This included the cam drive for the spectrometer wavelength scanning mechanism, the front plate and the detector housing.

Finally, the contract was revised to package the four spectrometers not adapted to the shuttle orbiter scanning platform together. These instruments, along with the two on the scanning platform, have been shipped to Air Force Geophysics Laboratory.

2.0 SPECTROMETERS - GENERAL INFORMATION

The seven spectrometers are all 1/8 meter Ebert-Fastie mount with an aperture ratio of f/5. These instruments all have a wavelength scanning, plane grating and a concave spherical mirror. The wavelengths, grating ruling and blaze wavelength are presented in Table I.

WAVELENGTH DATA

SERIAL NO.	MODEL NO.	WAVELENGTH RANGE	GRATING RULING	GRATING BLAZE
020	17-193	1150 - 1900	3600	1300
026	15-224	1800 - 3200	2400	3000
027	15-224	5000 - 8500	1200	5000
031	15-224	1600 - 2900	3600	2400
032	15-224	2700 - 5000	1200	5000
033	15-224	1100 - 1900	3600	1216
034	15-224	1100 - 1900	3600	1216

TABLE I

The wavelength scanning drive of the grating is a sinusoidal drive implemented with a cam. The single revolution cam has a linear rise and a cycloidal fly-back so that scanning is always done in the same direction. The prototype instrument has an internal cam and is pictured in FIGURE 2. The six subsequent flight instruments incorporate an external cam drive. The revised assembly of the flight spectrometers is shown in FIGURE 3. This change was necessitated by the manufacture of the photomultiplier tube (PMT) discontinuing production of the detector used on the prototype. The replacement PMT is larger in diameter and interferes with the original cam motor mounting arrangement.

The same telescope is used with all of the spectrometers. FIGURE 4 displays the telescope assembly. Notice the alignment mirror assembly used to co-align several instruments to the structure is illustrated here also. A dust cover is incorporated into each telescope to protect the optics when the instrument is not in use.

3.0 FUNCTIONAL CHARACTERISTICS

3.1 Spectral Bandwidth

The spectrometer bandwidth, as defined by the grating ruling and slit size, range from 4.6 to 5.7 Angstroms for the various instruments.

3.2 Field of View

The maximum field of view of the spectrometer/telescope system is 0.05 degrees (0.00087 radians). This was not measured precisely, but was calculated from design dimensions and mechanical and optical tolerances.

The prototype unit was mounted on a scanning platform which oscillated through a 20 degree (0.35 radians) arc. This field-of-view scanning platform was cam driven. The mechanical arrangement is illustrated in FIGURE 5.

A pair of flight instruments were mounted on a scanning platform which is driven by means of a stepper motor through a worm and wheel gear train. This is pictured in FIGURE 6. The field-of-view scan for this mechanism is 75 degrees (1.31 radians).

3.3 Shutter

A torque motor operated entrance shutter is mounted on each spectrometer. This can be used to make dark count readings with the detector. It may also be connected to the solar sensor to protect the detector against unintentional overload from viewing too near the sun.

3.4 Dust Cover

The dust cover pictured in FIGURE 7 is driven by a stepper motor. Fiducial readers at each end of the motion indicate when the cover is fully open or fully closed. The dust cover must be completely open in order for the instrument to operate properly.

3.5 Pump out Baffles

Pump out ports to allow equallization of pressure are provided on both the telescope and the spectrometer. These openings are baffled to prevent stray light from entering. The pump out ports also allow nitrogen to escape when the instrument is being purged.

3.6 Purge Ports

Purge ports are provided to flood the telescope and spectrometer with dry nitrogen prior to flight. This is to keep dirt and humidity from degrading the optics. The plumbing for the purge is equipted with a quick-connect fitting.

3.7 Solar Sensor

Solar sensors were provided to protect the spectrometer detector from damage by exposure to too bright a source. This sensor, shown in FIGURE 8, has a 20 degree (0.35 radians) field of view. The single solar sensor is capable of protecting several co-aligned instruments.

4.0 OPTICAL DESIGN

4.1 General

The basic design of the spectrometer is an f/5 Ebert-Fastie system, with a focal length of 1/8 Meter. Straight slits were used since the slit height was short, still allowing good image properties to obtain 5 Angstorm resolution. Detailed characteristics of the optical system are shown in FIGURE 9, the system optical layout, and FIGURE 10, the optical layout of the spectrometer.

4.2 Telescope Mirror

The telescope uses an off-axis parabolic first surface mirror. The mirror was fabricated from Cervit without the optional center hole shown in FIGURE 11. The rear of the blank was cut away to reduce weight. The parabolic surface is aluminized and coated with MgF_2.

The mirror is constrained by eight (8) non-metallic tipped screws around the edge. Two screws on each side permit the necessary adjustments along the X and Y axis as well as rotation about the Z axis. Longitudinally the mirror is clamped at three (3) points between non-metallic pads.

4.3 Ebert Mirror

The Ebert mirror was made from precision annealed Cervit material. No weight relief was performed on the mirror in order to eliminate distortion when mounted in the mirror cell. The radius of curvature was 250 mm $\pm$ 1 mm. Three (3) flats were ground into the front surface of the mirror to provide mounting points, at which places non-metallic mounting pads contact the mirror. The mirror is constrained around the sides and back by non-metallic tipped screws.

The mirrors were aluminized and overcoated with either MgF_2 or SiO, whichever was appropriate to the wavelength being investigated. The Ebert mirror is shown in FIGURE 12.

4.4 Diffraction Grating

The replica diffraction gratings used were ruled and blazed as indicated in TABLE I. The grating had a ruled area of 27.5 mm square. It was aluminized and overcoated with either MgF_2 or SiO, as appropriate for maximum reflectivity in the working range. The grating blank is shown in FIGURE 13.

4.5 SLITS

The precision machined slits used in the spectrometers are cut on standard punched blanks. Separate blanks are used to produce the height mask and width slits. They are illustrated in FIGURES 14 and 15, respectively. This two piece construction permits varying the height and width independently.

The slits are accurately positioned in the front plate by means of the matching flat. The slit widths on the instruments varied from 0.800 to 0.079 mm with a slit height of either 6.0 or 7.0 mm.

The slits are assembled with the height mask ahead of the width slit. The chamfer on the height mask is in the direction of the light rays and width slit is reversed. Notice that this results in the entrance slit being installed in the faceplate opposite to the exit slit.

4.6 Telescope Baffles

All of the interior surfaces of the telescope are baffled to eliminate unwanted stray light. Furthermore, a sun shade at the entrance to the telescope provides additional rejection of off-axis rays.

4.7 Spectrometer Baffles

Internal baffles were used in the spectrometer to reduce unwanted stray light from passing through the optical system and entering the detector. These baffles were placed in front of the Ebert mirror. The grating was also fitted with a mask to keep light from reaching unused portions of the blank.

5.0 OPTICAL ALIGNMENT

5.1 General

This optical alignment procedure is the standard procedure for aligning a RSI 1/8 Meter Ebert Spectrometer. The principal elements of the spectrometer requiring alignment are:

a) the Ebert mirror,

b) the grating,

c) the spectrometer housing,

d) the wavelength scanning mechanism.

5.2 Ebert Mirror

The mirror cell cavity is approximately 4 mm larger than the mirror to allow for lateral movement during optical adjustment. The mirror is held laterally by six (6) set screws with non-metallic tips. Four (4) screws are along the length of the mirror and two (2) screws are at the ends. All of the screws oppose one another.

Place the mirror in the housing, face up, and center it by adjusting the six (6) set screws against the side and ends of the mirror, but do not tighten.

Place the mirror mask with buttons over the face of the mirror, noting the alignment of the three (3) buttons with the three (3) corresponding locating pads on the mirror. Move the mirror face towards the buttons by adjusting the three (3) opposing set screws until the mirror just touches the buttons.

The mirror and cell assembly can now be installed onto the main housing with the adjusting shim between the mirror mask and the main housing. Clamp the mirror cell in place with 10 #2-56 x 1/2" long socket head cap screws and tighten to 1.4 inch-pounds torque.

A preliminary check of the Ebert mirror alignment can be made by attaching a flat test mirror to the inner surface of the front plate keeping the face of the test mirror parallel to the axis of rotation of the grating.

With the test mirror attached to the front plate, install the front plate onto the opposite end of the main housing. Focus a 4X power eye loop onto the exit slit opening with the slits installed. Looking through the loop, the image of the entrance slit should be visible. If the slit image of the slit openings are not in line, adjust the three (3) set screws at the rear of the mirror until the proper slit opening alignment is achieved. Remember, this is a preliminary mirror alignment and will be finalized after the grating is installed and aligned.

5.3 Grating

The box which holds the grating is designed in the same fashion as the mirror cell. Set screws with non-metallic tips at the sides and back and Kel-F pads at the front support the grating.

Paying attention to the relationship of the grating box with respect to the entrance and exit slit position, place the grating into the box face up with the blaze arrow pointing toward the exit slit. Center the grating and keep the ruled lines of the grating parallel with the grating box cavity walls by tightening or loosening the set screws at the sides of the grating.

Place the grating mask with buttons over the face of the grating, noting the alignment of the three (3) buttons with the three (3) adjusting set screws at the back of the grating which must oppose the buttons. Tighten the three (3) set screws at the back of the grating until the face of the grating just touches the face of the locating buttons.

5.4 HOUSING

With the grating and mirror contained loosely in their respective holders, the final alignment procedure can be carried out as follows:

a) The mirror cell should be firmly secured to the main housing.

b) The front plate with the rotatable grating box assembled to it can be clamped to the main housing with only a few screws because it must be removed each time a grating adjustment is required.

c) Install alignment slits that have about a 1/2 mm width opening and a cross hair through the center of the slit.

d) Attach an arm to the grating box journal so that the rotation or "fanning" of the grating through the spectrum wavelength can be controlled.

e) Focus the 4X power eye loop on the exit slit.

f) Illuminate the entrance slit light path with a mercury lamp or some other suitable light source that will give a visible spectra.

g) You should now be able to see the entrance slit with the cross hair and the reflected image of the exit slit and its cross hair through the eye loop, by fanning the grating back and forth through both the positive and negative spectrum.

h) While fanning the grating from negative to positive, observe if the image of the cross hair is constantly in line with the stationary cross hair.

2) If the image of the cross hair remains straight while fanning the grating but is above or below the stationary cross hair, then the tilt of the grating must be corrected by adjusting the three (3) set screws at the back of the grating. This condition can be improved by adjusting the tilt of the Ebert mirror after optimizing the grating's parallelism with the grating box axis of rotation.

i) When the alignment is satisfactory, the set screws locating the mirror and grating are part of the clamping arrangement and must be tightened. Tighten the set screws holding the grating to 1.4 inch-puonds torque, and those holding the mirror to 3.6 inch-pounds.

5.5 Focus

With the alignment complete, the focus must be checked with the use of a 10X power microscope. Install the actual slits at their respective positions. Focus the microscope on the exit slit, keeping it in line with the spectrometer light path. Illuminate the entrance slit opening with a mercury lamp. Set the grating so that one of the edges of the entrance slit is imaged into the microscope with the spectrum set at the green line, focus the microscope on the imaged slit. The distance, if any, that the microscope must be moved in order to focus on the exit slit then focus on the entrance slit image is the distance that the optics are out of focus. This focusing error is corrected by either increasing or decreasing the thickness of the shim between the mirror mask and the main housing one half of that difference.

The next procedure for the aligning the spectrometer would be to set the wavelength in conjunction with the cam's designated wavelength scanning motion and pin the grating arm. Because a telescope type collimator is to be installed on the front plate we must leave the grating box free to rotate in order to check the alignment of the telescope optics with the spectrometer optics.

5.6 Telescope Optics Alignment

To facilitate the optical alignment of the telescope, the following items are required:

a) Six foot optical bench.

b) Horizontal and vertical slides mutually perpendicular to the optical bench centerline.

c) Mounting device which permits rotation of the instrument about the vertical axis perpendicular to the optical bench centerline.

d) Helium-Neon Class II Laser.

e) A set of alignment plates that mount to the telescope, with a 1/16" aperture at the center.

The laser must be on slides that allow it to be moved vertically and horizontally while maintaining the parallelism with the longitudinal axis of the optical bench. Mount the telescope body securely to the optical bench along with an alignment plate fastened to the front and rear of the telescope. Those alignment plates should have a pilot hole through them that represents the center of the telescope. Adjust the telescope and/or the laser until the laser beam is projected through each alignment plate. After this has been done, the telescope and the laser are now in line with the optical bench axis.

Remove the back or mirror end alignment plate and install the mirror housing along with the mirror which is located by three (3) Kel-F buttons at the face of the mirror, three (3) non-metallic tipped set screws at the rear of the mirror that oppose the buttons and eight (8) soft tipped set screws at its sides which also oppose one another. With the mirror in place, the laser beam should now be passing through the center pinhole in the front alignment plate onto the mirror and reflecting back through a second pinhole in the alignment plate. The set screws at the back of the mirror must be adjusted until the beam passes through the second alignment pinhole. Next, scan the laser to the outermost reflecting areas of the telescope's parabolic mirror and adjust the mirror so that the reflected laser beam passes through the center pinhole of the second alignment plate at the slit position

Once this is accomplished, the set screws can be tightened to 3.6 inch-pounds torque, while keeping the alignment in tact. Without disturbing the positioning of the telescope on the optical bench, mount the spectrometer onto the telescope. Set the grating at zero (0) order and pass the laser beam through the spectrometer and out the exit slit. If the beam does not come out the exit slit, the alignment procedure must be repeated.

5.7 Wavelength Setting

When the optical components are in proper alignment and focus and securely mounted, the mechanism that defines the spectrometer's specific wavelength scanning range can be set. To set the grating to the proper angle that will define the wavelengths to be scanned requires special tooling. This tooling allows the grating to be set at zero (0) order and then very precisely rotated to a predetermined angle which defines the minimum wavelength setting. With the grating at the predetermined angle and the cam follower pin set at the lowest point on the cam, the grating arm can be secured to the grating box journal. The scanning range is fixed by the rise of the cam. If a different wavelength scanning range is required, a new cam must be installed. The scanning mechanism has a fine adjustment that allows the experimenter to tune the wavelength setting to the specific photomultiplier tube being used.

6.0 ELECTRONICS INTRODUCTION

The electronic documents are described, the state of the instrument at completion of contract is detailed, and suggestions for preparation of a specific flight are given.

6.1 Documentation

Two major documents are included which cover the electronics for these instruments, (1) Electronics Manual Flight Instrument HUP2 and (2) Calibration and Testing System. The first covers all the electronics for the flight instrument and details the microprocessor software. The second describes the calibration and testing system which is used to operate the flight instrument on the ground during calibration and check out.

Copies of all documents are provided om MS-DOS disk to make editing easier.

6.1.1 Flight Instrument Electronics Manual

The Flight Instrument Electronics Manual first describes each piece of electronic hardware and gives critical parameters for its operation. Pinouts and physical size are not covered here but are available on the appropriate mechanical drawings

The second section describes the software for the flight controller. A general description and design philosophy are given and then details of the flight program are layed out.

The third section explains in a step-by-step fashion how to develop software for the controller and burn PROMs. This makes adapting the software to specific flight requirements a straightforward effort.

Also included are a variety of appendices giving the Memory and I/O Map, Data Word Bit Identification, Forth Core Words, 8085 Assembler Words, and References. Listing for three versions of the flight software are attached at the end.

6.1.2 Calibration and Testing System

The Calibration and Testing System Manual describes the use of a custom RSI CTS card in an IBM compatible computer to calibrate and test the flight instrument. Details of the hardware are given with available options and then a description of the driving software.

A detailed explanation is given for adapting the CTS system to various flight configurations.

6.2 State of Electronics

The completion of the electronic components and proving of software was stopped when the last flight opportunity was lost. It did not seem prudent to spend money completing portions of the system which would have to be altered to suit a new flight. When a flight is defined, the existing hardware and software can be modified and completed to suit the particulars of that flight in a relatively short time.

6.2.1 Flight Software

The flight software is complete and has been tested on a breadboard system. Some modification will be needed to match a new flight configuration but most of this will be editing of the existing program. Debugging in Forth is both straightforward and fast.

6.2.2 CTS Software

The CTS software will have to be adjusted to suit the new data configuration but this system is very adaptable in this regard.

6.3 Flight Preparations

This flight instrument can be adapted for use on a wide variety of spacecraft and even land or marine systems. This involves first defining the components to be included and then adjusting the hardware and software to suit the new system.

6.3.1 Components

This instrument may be used as a group of one, two, or four spectrometers which maybe mounted on a scanning platform. Each instrument will have a detector, shutter, dust cover, wavelength scan motor, fiducials, and microporcessor controller. Each may have a mercury lamp and a sun sensor.

The software for the individual controllers can be easily edited for the specific components included.

6.3.2 Power

This instrument uses +28 VDC power and +5 VDC power. The +28 VDC is usually provided directly from the spacecraft power supply and runs the motors, integrated detectors, and mercury lamps. A small input filter is included in the instrument.

If the available voltage is substantially different from +28 VDC then a DC to DC converter will be needed.

The electronics runs off +5 VDC which must be provided by a DC to DC converter. Flight power supplies of this type are available from several suppliers.

The Flight Instrument Manual contains the information needed to size these power supplies.

6.3.3 Communications

The present design of the instrument expects a Data Bit Clock and a Data Enable Signal from the spacecraft and will provide the data in the form of serial data. This system may be easily adapted to a new flight environment which uses a different number of instruments, different clock rates, and even different signal levels.

The design of the Spacecraft Interface Module may have to be adjusted for the time of signal. The present design assumes single TTL compatible signals, but it will not be difficult to adjust the design for differentially driven signals, or CMOS level signals.

The number and meaning of the data bits may be adjusted in hardware and is described in detail in Appendix B. First, edit Appendix B to suit the new conditions and then adjust the software appropriately.

If the new system communicates over a RS-232 or RS-422 system, these can be accommodated by using the RS-232 development line designed into the Controller. Appropriate level shifters will be needed in the Spacecraft Interface Module. Software modifications will also be needed but this will be limited to a single screen.

6.3.4 Space

The space available for the new configuration will determine the physical layout of the system and may dictate the required shape of the system. The present layout for the microporcessor board has all the connectors along one side. This is convenient for cabling in some applications but difficult in others. The development connectors are also mixed in with the flight connectors and may be hard to reach if cables are in the way.

Available space also clearly has an affect on cabling.

APPENDIX A

AFGL-801A HUP

INTRODUCTION

The AFGL-801A Horizon Ultraviolet Program Experiment consists of a spectrometer which scans from 1100 to 1900 Angstroms, a platform which scans through an angle of 20 degrees, and various support modules. This document will cover the seven (7) various support modules' contents and operation. The following modules are included.

1. +5 V Power Supply
2. +/-15 V Power Supply
3. 8085 Microprocessor Control
4. Stepper Motor Power Control
5. Integrated Detector
6. Solar Sensor
7. Mercury Vapor Calibration Lamp

Spacecraft interface is provided via flight connectors J1, J2 and J3. There is also an AGE connector for GSE communication via the HP-85 and its RS-232C interface Model 82939A. The following enclosed schematics should be referred to for full understanding of the 801A experiment.

+5V Power Supply Input Stage	
+/-15V Power Supply	
Microprocessor Controller Circuitry	800-193-0-7
Stepper Motor Power Module	480-193-0-4
Integrated Detector	230-215-0-2

Solar Sensor	680-215-0-4
Hg Calibration Lamp	170-193-0-2
+28V Power Filter	442-193-0-1
AFGL-801A Wiring Diagram	4-193-700-1
Stitch Weld Board-Stepper Motor Driver	480-193-40-1
Stitch Weld Board-Microprocessor Cont.	800-193-40-1

All connectors are presently labelled with nylon cable tags so that identification should be no problem. The experiment is powered by +25 to +33 VDC at approximately 1 to 1.5A. The major power sinks are the five (5) motors (spectrometer, platform, dust cover, and latch stepper motors and the slit shutter torque motor) and the +5V power supply (Abbott Transistor Labs Model C5DH2.5). RFI shielding is provided on the majority of the wiring harness and all electronics boxes and connectors are case grounded to the support structure. During EMC testing, it was found that a power filter was necessary to keep the conducted emissions in specifications. A power filter was therefore included in series with the input power connector. This power filter is an inductor-capacitor pi type. Electro-optical fiducials are provided with all the stepper motor systems for limit or mark information to the microprocessor.

Other features are a slit shutter which is used for dark counts, a solar sensor which turns off the integrated detector high voltage and closes the slit shutter in the event of the sun being within a 10 degree field of view, a mercury vapor calibration source for wavelength calibration of the spectrometer, a dust cover to prevent contamination of the system optics, and a latch for securing the instrument during ascent and descent.

SECTION I

POWER SUPPLIES

The power input to the experiment is defined in the ICD as 25.0 to 33.0 VDC not including ripple. The experiment has two (2) power systems to convert this nominal 28V power to 5V and to +/-15 VDC. The 28V to 5V DC/DC converter is manufactured by Abbott Transistor Laboratories, Inc. It is Model C5DH2.5 and its input circuitry is included with the schematics. This information is proprietary so that only the input capacitor value was provided by Abbott Transistor. The 28V to +/-15V DC/DC converter is RSI Model 441-193 and is capable of providing well regulated +/-15 VDC up to a 400 ma. load. Both of these supplies are hermetically sealed.

It should be noted that the power is routed from spacecraft interface connector J1 to the 5V supply whose housing acts as a junction box for re-routing the +28 VDC as well as providing the 5 VDC. There is a 1 mH inductor contained inside the housing in series with the +28 V power lines for inrush current limiting.

INTEGRATED DETECTOR

The integrated detector package is a modified version of RSI Model 230-210 and is designated with a new model number, RSI Model 230-215. This modification was isolating the three (3) grounds - case, signal and 28V return. All other electronics remain the same.

The integrated detector contains a PMT, pulse amplifier discriminator and a high voltage power supply. The photomultiplier tube is an EMR Type 510-G and is sensitive in the wavelength region 1150 to 2200 Angstroms. High voltage is adjustable over the range 1500 volts to 3000 volts with a 1000:1 HV monitor output. There is a thermistor encapsulated near the front face of the PMT for temperature monitoring. The unit is encapsulated for operation at any pressure from one atmosphere to space vacuum.

MERCURY CALIBRATION SOURCE

Incorporated in the experiment is a calibration source consisting of a miniature Hg vapor lamp and a power supply. The lamp is custom constructed for RSI by Ultraviolet Products, manufacturers of the well-known Pen-Ray lamps. It is 4mm in diameter in order to fit between the light baffle vanes in the telescope. The power supply produces the breakdown voltage and also the current for rapid turn-on and warm-up. It is controlled by the microprocessor via a relay in the power module.

It must be realized that the location of the lamp does not allow direct irradiation of the spectrometer slit and thus the grating. The method employed for wavelength calibration depends on the scattering of light from the back of the dust cover which is left as bare aluminum. The lamp illuminates the telescope mirror, which, in turn, illuminates the back of the dust cover. The dust cover scatters light back to the mirror and into the spectrometer system.

SOLAR PROTECT SENSOR

The solar protect sensor consists of a phototransistor and trip circuitry for indicating the presence of a bright source in the field of view of 10 degrees (+/-5 degrees about the direction of the telescope), the bright source being the sun in this case. Outputs of the sensor are a Sun Presence signal and the Intensity of the source. (A Threshold Level is output for initial set-up.) The Sun Presence signal is a discrete level output signifying whether the source intensity is or is not above the trip point set by the Threshold Adjust potentiometer. The other signals are analog in nature.

SECTION II

EXPERIMENT CONTROL THEORY OF OPERATION

The brain of the AFGL-801A experiment is an 8085A Microprocessor, U1. The processor is programmed to respond to the eight (8) commands, check various status information, provide housekeeping, step the appropriate motors and output data and errors to either the spacecraft interface or the HP-85 RS-232C interface. It conserves power in the experiment by powering up or down the various motors at the right times. The detailed discussion below will be clearer if the microprocessor controller and the power module schematics are reviewed.

The 8085A microprocessor (U1) is linked to the outside world through its support chips, its interrupts and serial ports. The support chips are one (1) 8755 EPROM/IO (U15), two (2) RAM/IO (U7 and U14), and one (1) 8251A UA/ART (U3). The 8205 U6) selects one of the four (4) chips attached to the microprocessor address/data bus.

The 8251A US/ART provides communication to the HP-85 RS-232C interface. The baud rate clock for the US/ART chip is provided by U37 and U4 dividing the processor clock (3.072MHz) by 5 and then 16 respectively, a total divisor of 80. This communication was configured to run at 2400 baud by using the divider included on the US/ART to divide by 16 again. This gives us:

$$3.072\text{MHz}/(5\times16\times16)=2400$$

Finally, the voltage levels are made RS-232C compatible using level shifters U35 and U36.

The other three (3) chips on the microprocessor bus are memory chips with I/O ports. The 8755 EPROM contains the stored program and the two (2) 8155's are used for scratch-pad memory, retaining the system status, previous data, fiducials, etc. The I/O ports on these chips comprise the most important link between the microprocessor and the experiment: through these ports, information is supplied to the 8085. (There is also some limited data supplied to the 8085 via its serial input port-(SID) and limited control via the serial output port-(SOD).) These ports provide the following support:

1. Input the data from the integrated detector's pulse counter.

2. Send motor pulses to the Stepper Motor Power Module to step the various motors.

3. Disable pulse input to the counters during a count latch.

4. Close the slit shutter for dark counts.

5. Output the data to the serial shift register.

6. Input data from the A/D converter.

7. Input command latch data.

8. Input fiducial data.

9. Control various multiplexors for data selection.

10. Override the solar sensor lockout.

11. Operate the calibration lamp.

12. Power up/down the different motors, i.e. operate the latch

relays.

I/O INTERFACING

The I/O interfacing will now be addressed by examining each port's operation. The ports are labelled PORT A1, PORT B1, PORT A2, PORT B2, PORT C2, PORT A3, PORT B3, PORT C3. To be included in this discussion are the two (2) timers on the 8155 chips, TIMER1 and TIMER2.

PORT A1 and PORT B1 are contained on the 8755 EPROM/IO chip (U15). PORT A1 is used for transferring the six (6) data bytes to the serial shift register U16 for output to the spacecraft. This port does not load the data into the shift register, but only presents it there, pending a load pulse from TIMER1 OUT. This timer is set up to send a pulse for every eight (8) pulses it receives. This pulse is shortened by the 54LS221 (u39) to keep data loading at the proper time (see Timing Diagram, Figure 16). After the byte of data has been loaded, the software updates the parallel byte to the next byte for output. In this manner, only one (1) parallel to serial shift register is necessary. The serial output line is buffered and routed to spacecraft interface connector J3 - Digital Signal Out. Serial shifting is timed to the Digital Shift Clock provided by the spacecraft. This signal also increments the timer so that on the eighth data bit being shifted out, another byte will be loaded. After the sixth byte has been output (and the Data Shift Clock discontinued), it is necessary for the processor to set the first byte loaded in before the next Enable Data Strobe. This is done via software and PORT C3, Bit 1. (Note: Bit designation goes from Bit 1 for least significant bit to Bit 8 for most significant bit in this discussion.)

PORT B1 is a multi-purpose input port used to input data from either the on-board A/D converter, the command latch, or the fiducial buffer. It basically has its own little bus with three (3) tri-state buffers (U22, U23 and U25) attached. The buffer selection is accomplished using bits 1 to 3 of PORT A3.

The first 8155 (U7) contains ports A2, B2 and C2. PORT A2 and PORT B2 are the input ports used to read the present data pulse count in the counter latches. PORT C2 is used for three (3) functions. Bits 3, 4, 5 and 6 are used to pulse the spectrometer, platform, dust cover, and latch motors, respectively. Bit 1 gates the data pulses into the data pulse counters, which are ripple counters, previous to the count being latched into U8 and U10. This prevents any counter-action (i.e. rippling) during latching. The pull up resistor, R58, is needed because the ports are configured as inputs on power up and R58 pulls the line high, not allowing pulses to the counters until the processor allows it. Bit 2 of PORT C2 is used to close the dark shutter under microprocessor control. TIMER1 is also on U7. Its function is to provide a pulse each time eight (8) Data Shift Clock pulse occur. This pulse triggers U39 to generate a shorter pulse to load the byte presently at the shift register's parallel input lines. The pulse also tells the microprocessor that the new byte is loaded for it to set up the next byte on the parallel lines. The timing diagram (Figure 16) shows how this all falls together. You will notice that PORT C3, Bit 1 is used in conjunction with TIMER1. This is needed in order to load the first byte into the shift register, anticipating the next data output sequence (see Figure 16).

The second 8155 (U14) contains PORT A3, B3 and C3. This chip is used primarily for multiplexor and latching relay control. PORT A3 Bits 1, 2 and 3 are used to select one of five (5) operations through 8205 (U2) one-of-eight decoder. These functions are listed in Table II. PORT A3, Bits 4 and 5 are used to select one of four (4) analog signals for presentation to the on-board A/D converter. The analog multiplexor is the LF11508 (U28) chip. The four (4) analog signals are defined in Table IV. PORT A3, Bit 6 is used in conjunction with command 4 in overriding the solar protect circuitry. PORT A3, Bit 7, is used to operate the calibration lamp relay in the Stepper Motor Power Module (Relay K5). PORT B3 is used solely for the power up/down latching relays K1-K4. The actual function of each bit is defined in Table I. As noted above, PORT C3, Bit 1 is used in loading the first byte into the serial shift register. PORT C3, Bits 2 and 3 are used to select one of four (0) digital lines to the microprocessor's SID input port through the 54LS193 (U19) one-of-four multiplexor. The four (4) inputs are defined in Table III. Note that PORT A3, Bit 8 and PORT C3, Bits 4, 5 and 6 are presently unused. TIMER2 on this chip is used when the HP-85 RS-232C interface is in operation. It generates an interrupt every 10 ms, simulating the spacecraft Enable Data Strobe signal. This 100 Hz signal is derived from the baud rate clock comprised of U37 and U4.

It should be noted that the addressing of any of the above and the US/ART is done by the addressing bus and the 8205 (U6) chip selector. The determination of which chip is selected depends on the configuration of the address bits A11 to A13. Thus, the address locations are defined as follows:

(U15)	8755	EPROM/IO	0000-07FF
(U7)	8155	RAM/IO	0800-08FF
(U14)	8155	RAM/IO	1000-10FF
(3)	8251A	US/ART	1800

TABLE I

Bit 1	Power Up Latch
Bit 2	Power Up Dust Cover
Bit 3	Power Up Platform
Bit 4	Power Up Spectrometer
Bit 5	Power Down Dust Cover
Bit 6	Power Down Latch
Bit 7	Power Down Platform
Bit 8	Power Down Spectrometer

Depending on whether the IO/M line from the 8085 is high or low determines whether the I/O ports or the memory section of these chips is selected, respectively. All chips on the processor bus are reset on power-up. This means all ports are configured as inputs, timers are disabled, and the US/ART is non-functional. It is the job of the microprocessor to initialize the US/ART and configure the ports correctly. Also, any user memory space should be cleared.

SUPPORT CIRCUITRY

The various building blocks which lend support to the microprocessor are outlined next.

1. Data collection.
2. Data output, T/M output.
3. A/D Converter.
4. Command Latch.
5. Fiducial input.
6. Relay Drivers
7. Solar Time-Out.
8. Multiplexers/Selectors.

All spacecraft interface signals are buffered by the AFGL-801A electronics package.

DATA COLLECTION

The integrated detector included in the AFGL-801A experiment outputs a TTL compatible pulse for each photon event which occurs on the photomultiplier face. This pulse is transmitted via a coaxial cable to the microprocessor electronics, where it is gated by PORT C2, Bit 1. From there, the data pulses are counted by a 16-bit binary counter, made up of two (2) 54LS393 chips (U9 and U11). Every 10ms an interrupt occurs and the microprocessor stops the counters and latches the data via the two (2) 54LS273's (U8 and U10), restarting the counters immediately after latching. Thus, at most, 5μ's are lost in the dead time. (The interrupt is generated either by the Enable Data Strobe or internally by TIMER2, depending on the interface - spacecraft or HP-85.) The counters are only reset at power on and thus continuously count in a ramp counter fashion. It is the microprocessor's duty to do the proper subtraction. Data is input to the microprocessor via the port's A2 and B2. Timing of these counters and latches is somewhat tricky in that it uses the ALE line of the 8085. See Figure 17 for a clearer picture of the timing involved here.

DATA OUTPUT

There are two (2) telemetry signals in the AFGL-801A experiment. The HV monitor of the integrated detector is a 1000:1 ratio of the actual high voltage on the photomultiplier tube (PMT). The HV monitor is buffered using one (1) op amp of a LM124 (U24). The other telemetry signal is a temperature monitor of the area near the face of the PMT. This is made up of a Fenwal thermister (Type GA51J11) and a +5 volt supply. Again, this voltage monitor is buffered with another op amp of the LM124

(U24). Note that there is a 5.1Kohm resistor in series with the output to satisfy the output impedance specification of the ICD.

The digital data output circuit is fairly complex. Its function is to take data stored in the microprocessor in parallel form and convert it to a serial bit stream, 48 bits long. This situation has to meet the interface specifications provided by the ICD concerning data synchronization. The complication arises from allowing the microprocessor to handle the data transfer and using only one (1) 8-bit shift register. The alternative would have been six (6) shift registers; the processor would need only to shove out the six (6) bytes of data and then idle. The decision was to have the microprocessor do the work and save on hardware. With this understanding, the data output circuitry will be explained. Refer to the Timing Diagram in Figure 16.

The timing of the digital data output is synchronized with the Enable Data Strobe and the Digital Shift Clock. To do this, the hardware utilizes the times on the 8155 chip (TIMER1). The input to this timer is the Digital Shift Clock OR-ed with PORT C3, Bit 1. The timer is set up to produce a negative going pulse for every eight (8) pulses presented to its input. Since there are eight (8) bits to be shifted out between each load, this synchronizes the parallel loads with the serial shift out. (At this time, it should be noted that the microprocessor's data transfer routine is synchronized by the Enable Data Strobe interrupting the microprocessor.) A few notes on Figure 16 are in order. There are tics above the Digital Shift Clock which indicate where the data are sampled by the spacecraft PCM. The tic marks above the Data Out line indicate where the data output of the shift register changes for the next bit output in pin 9 of U16. The Ø and 1's in this line show what value would be sampled by the spacecraft. This line also demonstrates that the data are stable at sample times.

Previous to the Enable Data Strobe going high, the first byte of data (the low order data count) is loaded into the shift register U16 in anticipation of the data transfer. When the Enable Data Strobe goes high, the microprocessor goes into its data output routine. The first byte has already been loaded, so its bit pattern is shifted out of the rising edge of the Digital Shift Clock. This clock also pulses the timer. When TIMER1 reaches an eight (8) count, it goes low until the ninth count, where it returns high. Thus, a pulse, one clock cycle long, is produced every eight (8) clock pulses. This pulse produced by TIMER1 is too long to be used as a load pulse of the shift register because the next clock is needed to make it go high. This would throw the data out of sync by one (1) bit each time a byte is loaded. Thus, a shorter pulse is needed and is provided by the 54LS221 one shot (U39), triggered by the falling edge of the TIMER1 Out Signal. This 1 μ s pulse is used for loading the shift register. In this fashion, all 48 bits (six bytes) are output to the spacecraft. However, in looking at the sample Data Out line in Figure 16, it shows a glitch can occur just before the next parallel load. This is due to the shift register outputting the serial input line of U16. It is of no consequence since the true data is loaded well before the sample time. (Look at the data output change tics to see when data is changed. The two (2) close tics indicate that the Serial Shift In Line has reached the output, due to the propogation delays in TIMER1 and the one-shot U39. It is immediately disposed of before data sampling occurs.)

One last comment will conclude the Data Output Section. Looking at Figure 16, it is apparent that, although 48 bits are shifted out, there is no provision for loading the next byte. This is due to the Data Shift Clock being turned off at the end of the transfer. Thus, PORT C3, Bit 1 is used to load the first byte of the next data transfer and also increment the timer/counter to the proper count. Thus, it is that the output data is always one transfer time behind real time. Also note that the output signal line to the spacecraft is buffered.

A/D CONVERTER

Included in the microprocessor module is an analog to digital converter (A/D converter) which is capable of converting up to eight (8) analog signals. There are presently three (3) signal which can be converted: The HV Monitor, the Temperature Monitor of the Integrated Detector, and the Intensity Monitor from the Solar Sensor.

The A/D Converter circuitry consists of a 1 of 8 analog multiplex or of (U28), a sample and hold (U26), the A/D converter (U27) and the tristate buffer (U25). The board which contains these analog components is a double-sided copper-clad printed circuit board to act as a well shielded ground plane board. This board is then rivited onto the digital board in the microprocessor module. It should be noted that there is only one point in common between the analog ground and the digital ground. This occurs just beside the A/D converter and is used to keep digital transients from destroying the accuracy of the converter.

The theory of operation is as follows:

The microprocessor selects one of four signal lines of the analog multiplexor via port A3, Bits 4 and 5. The LF 11508 chip (U28) decodes this information and passes the appropriate signal to the sample and hold (U26).

The sample and hold and the A/D work in tanden. For the AD571 (U27), the sample and hold amplifies (SHA) AD582H (U26) serves as a high impedance buffer. The control signals are arranged so that when the control line goes low, U26 is put into a "hold" mode, and the A/D U27 will begin its conversion cycle. The DATA READY (DR) line is fed back to the other side of the control date so that U26 cannot come out of "hold" mode during the conversion cycle. At the end of the conversion cycle, DATA READY goes low, automatically placing U26 back into the sample mode.

The processor initializes a conversion cycle by selection 03 of the 1 of 8 decoder (U2). Since the A/D is used in pulse control mode, the processor, after starting the conversion, then gates the tri-state buffer (U25) and awaits a signal that the conversion is complete.

The DATA READY line signals the microprocessor through its SID port. The Processor then inputs the converted data via the tri-state buffer U25 and PORT B1.

COMMAND LATCH

Commands are brought into the 801A experiment via connector J2. These signals are described in detail in the ICD. In brief, they consist of +25 to +33 Vdc pulses of $\sim$ 106 ms duration. The input circuitry used to latch these various pulses is zener diode CR11-CR18, opto-couplers OC1-OC8, and R-S flip-flops U29 and U31. The zener diodes have a breakdown of 15V, thus, no current flows in the LED portion of the opto-couplers until the pulse is greater than 15V. This guards against noise triggering the R-S latches. After the pulse exceeds 15V, the opto-coupler turns on and sets the appropriate flip-flop. This information is latched until the microprocessor sends out a reset pulse via 04 of the 1 of 8 decoder U2. When the proper time occurs in the software, the latch commands are loaded from U29 and U31, and then a reset pulse sent (a negative going pulse). The software then responds to the command information.

FIDUCIAL INPUT

There are eight (8) fiducial monitors on the 801A experiment. (A fiducial is a marker indicating a given mechanic configuration of the experiment, e.g., platform at stowed position.) These fiducials are:

1. Latch Closed

2. Latch Open

3. Latch Attitude

4. Dust Cover Closed

5. Dist Cover Open

6. Platform Cam

7. Platform Stepper Motor

8. Spectrometer

The latch fiducials (1, 2 and 3 above) give information concerning the position of the latch pin. Latch Attitude refers to the angle of the platform. This fiducial gives an indication that it is okay to close the latch. Items 4 and 5 are self-explanatory. The platform has a fiducial indicating each complete cam rotation (6) and thus, a complete platform cycle and another fiducial indicating each rotation of the stepper motor driving the platform (7). The spectrometer's fiducial indicates the start of each wavelength scan. A typical fiducial reader and the spectrometer fiducial are shown on the schematic.

This information from the fiducials is buffered via U22 and loaded into the microprocessor under software request through PORT B1.

RELAY DRIVERS

There are six (6) relays used in controlling the experiment. These include four (4) for power up/down of the motors and two (2) for on/off control of the Hg lamp and the integrated detector.

The power up/down relays are the latching type. They are controlled via PORT B2, its associated buffers (U17/4-U17/8 and U38/5-U38/8), and drive transistors (T2-5, T7-10). The microprocessor must send a pulse of 5 ms minimum to assure latch up of the chosen relay, after which the pulse may be removed.

The calibration source is turned on via PORT A3, Bit 7, buffer U17/8, and transistor T1. The integrated detector may be turned off via PORT A3, Bit 6, through U13/3, and transistor T6. These relays are not latching, so a continuous level must be present for the duration of the functions. (There is an alternate manner that the detector is turned off, discussed in the next seciton, SOLAR TIME-OUT.)

SOLAR TIME-OUT

The solar time-out feature is a circuit used to protect the integrated detector from damage due to excessive light levels (namely the sun). Attached to the experiment, near the entrance to the telescope, is a Solar Sensor (RSI Model 680-215). This sensor outputs two (2) signals - an intensity signal which goes to the A/D multiplexor, and a discrete level signal indicating whether the intensity is above or below the trip point set via a threshold adjust on the solar sensor. (See Schematic 680-215-0-04.)

This discrete Sun Presence signal is used to trip the HV Lockout circuitry, comprised of U30, U32, U33, U34, and various gates. Note that the signal is active low, i.e. Sun Prescence = 0 implies that the light level is too high. This signal is filtered to avoid any glitches triggering the time-out. It is then applied to the time-out circuits, the relay driver for the integrated detector, and the driver for the slit shutter torque motor. Thus, as long as Sun Prescence is low, the shutter is closed and the HV of the detector turned off.

On the rising edge of the Sun Presence signal, the one (1) minute time-out starts. (Sun Presence going high indicates light levels are okay.) This time-out circuit is used to assure that the HV is not repetitively cycled at a rapid rate; for example, due to a spacecraft spin or oscillation, which would flash the sun across the experiment time and again. If the Sun Presence should indicate the sun again before the one (1) minute has elapsed, the time-out is reset and awaits the disappearance of the sun. The cycle is then completed after one (1) minute with no sun by turning the HV on and opening the slit shutter. Total time out is t sun + one (1) minute.

A provision has been made to override the solar protect circuits via Command 4. This command is hardwired directly into the shut-off gates U13/3 and U13/4. Upon acknowledgement of Command 4, the microprocessor must set Bit 6 of PORT A3 before resetting the command latch.

Upon power-up, this time-out circuitry will be in an indeterminate state, i.e., the counters U30 and U34 will be set to an unknown count. Count will proceed until Bit Q2A of U34 goes high, resetting the timer U33. Thus, up to one (1) minute may elapse upon power up before the HV on the detector will come on and the experiment function. This should not be an inconvenience because the latch and dust cover are opening during this time.

MULTIPLEXORS/SELECTORS

There are several functions provided by the various multiplexors, selectors and decoders in the microprocessor module. These will be presented in sequential order.

U2 is a one-of-eight decoder 8205. Its function is to decode the three (3) least significant bits of PORT A3 to select one of eight functions as shown below:

TABLE II

A3	A2	A1	FUNCTION
0	0	0	Select Command Buffer
0	0	1	Select Fiducial Buffer
0	1	0	Select A/D Buffer
0	1	1	A/D Start Conversion
1	0	0	Reset Command Latch
1	0	1	Unused
1	1	0	Unused
1	1	1	Unused

The 8205 one-of-eight decoder U6 is used as a chip select controller, decoding the address bits A11 - A13.

U19 is a four (4) input multiplexer used to select various digital signals for presentation to the serial input data (SID) ports of the 8085. The four (4) signals and their address code are shown below:

TABLE III

S1	S0	SIGNAL

Ø	Ø	TIMER1 Out
Ø	1	A/D Data Ready
1	Ø	Optical Test Sensor
1	1	Solar Lockout Status

A brief definition of these signals is in order. TIMER1 Out is used to indicate when eight (8) bits of data have been shifted out through the shift register U16. Data Ready was explained in the A/D CONVERTER Section indicating when data may be accessed. The Optical Test Sensor signal shows when Hg calibration lamp is on. The Solar Lockout Status Signal indicates if the lockout circuits have shut down the high voltage and closed the slit shutter.

The last multiplexer is the LF11508 (U28) of the analog conversion section. Although it is capable of selecting eight (8) channels, only three (3) analog signals are used. The breakdown is:

TABLE IV

A2	A1	AØ	ANALOG SIGNAL
Ø	Ø	Ø	Solar Sensor Intensity
Ø	Ø	1	Detector HV Monitor
Ø	1	Ø	Detector Temperature Monitor
Ø	1	1	Unused

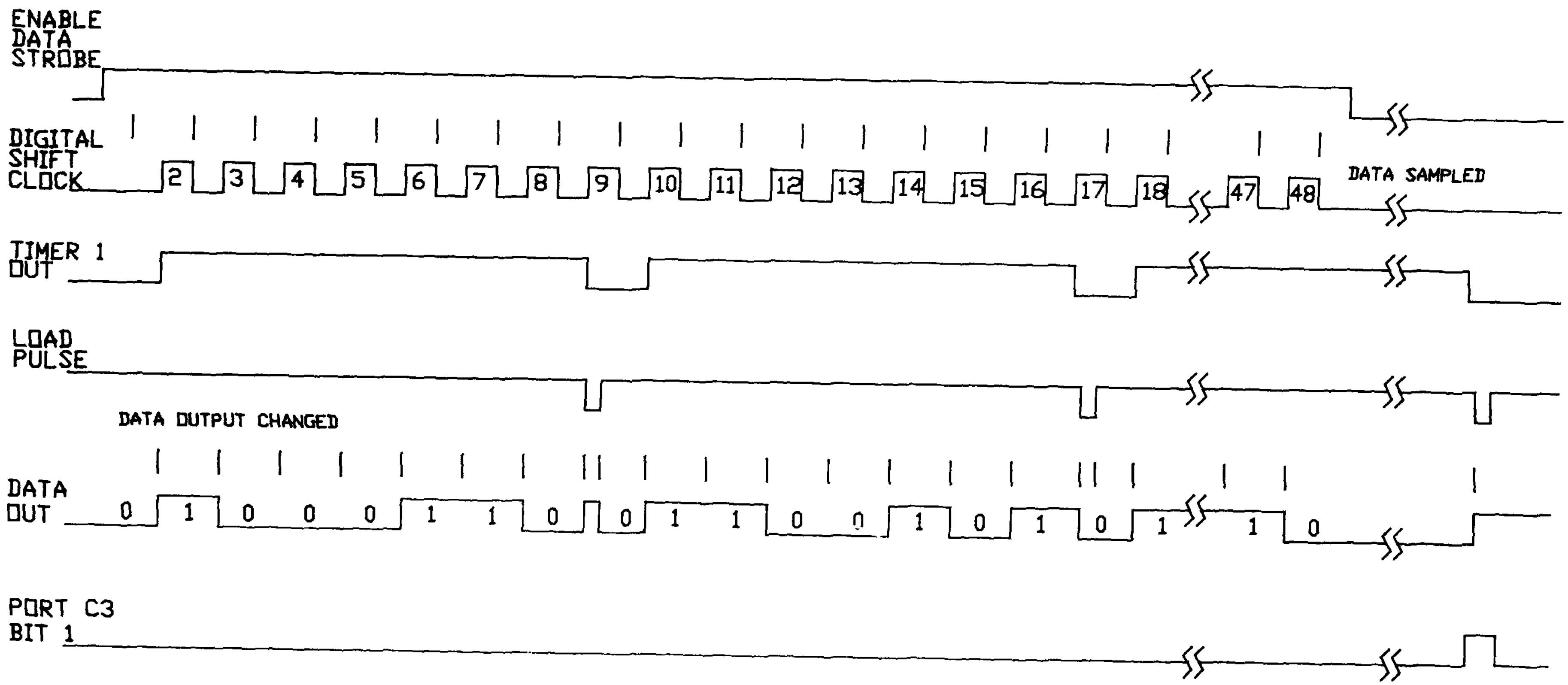


FIG 16
DATA OUTPUT CIRCUITRY TIMING

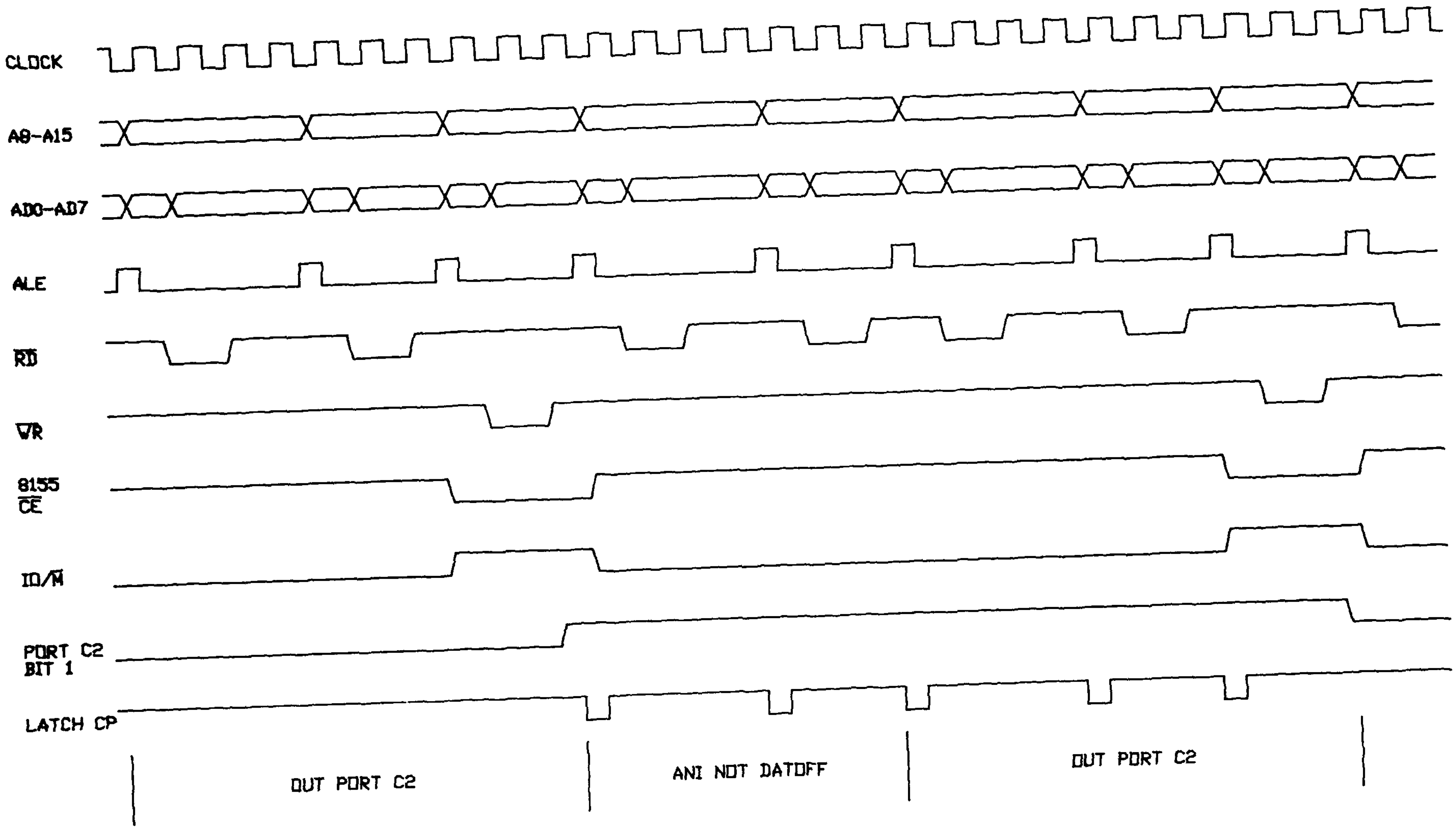


FIGURE 17

DATA LATCH CIRCUITRY TIMING

APPENDIX B

Test Plan No. RSI 79-C-0044-001 Prepared July 31, 1980

Acceptance Test Procedure

Horizon UV 801 Experiment

Prototype Spectrometer and GSE

AIR FORCE GEOPHYSICS LABORATORY
Hanscom Air Force Base
Massachusetts 01731

RESEARCH SUPPORT INSTRUMENTS, INC.
10610 Beaver Dam Road
Cockeysville, Maryland 21030

ACCEPTANCE TEST PROCEDURE
FOR VERIFICATION OF OPERATION
HORIZON UV 801 PROTOTYPE
SPECTROMETER AND GSE

I. Equipment Required

A. Prototype Spectrometer, Model 17-193-0-1, S/N 020

B. Ground Support Equipment

1. Terminal, Hazeltine 1420
2. Printer, Centronics 730
3. Data Processor, SE 269

C. Interface Cables

D. Electronic Equipment

1. Oscilloscope-Tektronix 541 or better with 10:1 Probe
2. Digital Voltmeter-Four digit +/- .05% or better
3. Square Wave Generator-+5, 0 to 1 MHZ with Attenuator
4. Counter- 10 MHZ, 40 MV sensitivity
5. Strip Chart Recorder-Brush 260 or equivalent

E. Optical Test Equipment

1. Vacuum chamber, 10 -6 Torr capability, ion pumped or LN2 trapped.
2. Source, hollow cathode or equivalent to provide spectral lines in 1200 Angstrom to 1800 Angstrom spectral region.
3. Red Laser.
4. Optical bench setup.
5. Hg test lamp.

II. Preliminary Preparation

A. Visual Mechanical Checkout

1. Assure that spectrometer is properly assembled, per drawing 17-193-0-1.
2. Assure that all interconnect cabling is properly connected.

B. Set Spectrometer Wavelength

1. Remove cover door on housing to allow access to grating arm.
2. Loosen screws in grating arm and remove, to allow grating to swing to zero order position.

housing.

4. Remove detector assembly, set up Hg test lamp over entrance slit. By eye, set grating to zero order position with micrometer at "0" reading. Be sure wavelength cam is set with follower pin at beginning of rise.
5. Move micrometer to number calculated from sine geometry for the beginning wavelength of the scan (.6027 inch for 1150 Angstroms).
6. Lock wavelength arm onto grating box with two (2) screws.
7. Remove setting arm and micrometer bracket, re-install access door.
8. Wavelength setting cannot be checked visually. Verification must be performed in a vacuum environment with proper sources.
9. Re-install detector and cabling.

C. Complete electronic checks to assure proper operation.

III. GSE Operations

The scanning spectrometer may be operated in one of four modes. These modes are:

SCAN The instrument gathers data across the entire measured spectrum with data accumulated at each step. The data accumulation period is controlled by SUMMATION PERIOD, a value selected during initialization.

STOP The instrument gathers data for an indefinite period at selected step values. The data accumulate period is controlled by SUMMATION PERIOD. The step values are selected and entered by the operator during initialization. Optionally, the operator may advance the instrument to the next selected STOP value or by a discrete number of steps using the video terminal for data entry.

PAUSE The instrument gathers data for a selected period of time at selected step values. The data accumulate period is controlled by SUMMATION PERIOD. The pause time controlled by PAUSE DURATION, a variable entered during initializations.

BAND The instrument gathers data between two (2) selected BAND values, as in the scan mode.

Data Initialization

Operating conditions for the instrument are entered using the video terminal (VDT) in response to the inquiry displayed. Up to nine (9) pages may be presented to the operator depending on mode selected.

Each page contains information to assist the operator in operating the instrument and establishing run-time parameters. These pages may be reselected for viewing by the operator without the need to re-enter data each time. The pages are as follows:

Page 1 Title Page

Page 2 Keyboard Operation

Page 3 Mode Select
Initialized to SCAN. The operator enters a number from 1 to 4 for Mode selection. At present, Playback Mode is not operational and the instrument will not accept an entry of 5.

Page 4 Summation Period
Initialized to 50 milliseconds. Used in all run modes.

Page 5 Select Pause, Stop or Band Waves
At power-on, initialized to all zeros. Unchanged from test to test. Note that entries are placed in the table in ascending order. Data may be entered at a specific table location by entering the following: Table location value (1 to 10) followed by a colon (:) followed by the data value (0 to 9999).

EXAMPLE: 2:105 followed by a RETURN.

A RETURN without data presents the next page.

This page is not displayed if SCAN is selected.

Page 6 Pulse Duration
Initialized to 1000 milliseconds. Displayed only in Pulse Mode.

Page 7 Data Acquisition
Allows selection of the printer and entry of time-of-year and run identification. The selections may be advanced using ne> or < keys (with shift key depressed). The RETURN key brings up the next page.

Page 8 Instrument Control

Allows turn on/off of the instrument, high voltage to the spectrometer, open the shutter (off) and the test lamp. These are initialized at power-on to off with the exception of the shutter which is closed (ON).

Note that the program is not at this time sufficiently intelligent to know that it cannot run in one of the four (4) modes with the instrument off. If the operator should inadvertently cause this to occur, power must be cycled off and on to recover control.

Page 9 Number of Test Runs

Default value of one. Run-time controls are displayed to assist the operator. One display, for printer control, is incorrect. The <O> should be <CTRL - O>. One updated feature, for STOP mode incremental step advance, is not displayed. To enable this function, depress the control and E keys together. <CTRL-E>

After the operator is satisfied that his entries are correct, a test is started by depressing the RETURN KEY at Page 9. The instrument will first synchronize to the start-of-scan position and begin to run. During a run, the instrument will output data to the analog and parallel digital outputs in accordance with parameters set up by the operator. Data to the CRT and to the printer are output on a one second basis. During a run, the control functions; CTRL-O for the printer, CTRL-S & S for KEYBOARD PAUSE, CTRL-C for run number clearing, CTRL-E for STOP mode data entry and S for STOP mode advance are active only when the CRT data is being refreshed.

At the outset of the test run, the instrument will display a summary page of the selected operating conditions. If the printer is selected, this data is also printed using a similar format.

At the end of a test run, the instrument will ask for continuation or termination. If continuation is selected, the instrument restarts at Page 1, reinitializing Mode to SCAN, Summation Period to 5Ø MSEC, Pause Duration to 1ØØ MSEC and number of test runs to 1.

If termination is selected, instrument power is turned off, the shutter is closed and the instrument enters an idle mode, ready for shutdown.

Note: A 32 Bit data accumulator is employed for DATA SUMMATION.

The Pause, Stop or Band values must be in an ascending sequential order. The table scan logic will terminate if a value in the table is smaller than the STEP count at the time of the search for a compare.

At the start of a run, if the printer has been selected for data output and if the printer is not on, the VDT display will be cleared and the cursor will be positioned near the lower part of the screen. No message will come up on the VDT. Turn the printer reset off and depress any key (the space bar is easiest) on the VDT to continue.

If the printer is on but in reset, a message, "PRINTER NOT ON!" will come up on the VDT. Turn the printer reset off and then depress any key on the VDT to continue.

DATA OUTPUTS

CRT - on a one second basis.

PRINTER - Also on a one second basis with a full line every four seconds.

DAC - At end of each SUMMATION PERIOD. If the spectrometer count is above 65,535, it is limited to 65,535.

PARALLEL PORT - At end of each SUMMATION PERIOD.

The sequence is as follows:

ID		
Ø1	t_o	STEP COUNT
1Ø	t_o + 2.5 MSEC	SPEC,COUNT,LSH
11	t_o + 7.5 MSEC	SPEC.COUNT,MSH, IF Summation period is > 5 MSEC else, this becomes LSH of SPEC. COUNT.

NOTES

<u>Printer Off</u>. After 'No. of test runs' Page (#9) no message. Clear page with cursor in lower left center. Turn printer on and depress any key.

<u>Printer On But in Reset</u>. Outputs message, 'Printer Not On!'. Turn Reset off and depress any key.

<u>Page 1</u>. Title Page

<u>Page 2</u>. Keyboard Operation (Return advances to Page 3.)

<u>Page 3</u>. Mode Select. Playback not enabled.

<u>Mode Descriptions</u>

1.	<u>SCAN</u>	Continuous
2.	<u>PAUSE</u>	Delay at value for given Pause Period.
3.	<u>STOP</u>	Like Pause except infinite Pause Period
4.	<u>BAND</u>	Scan from value to next higher value

<u>Page 4</u> Summation Period
<u>All Modes</u>, 5 to 10,000 MSEC

<u>Page 5</u> Select Pause Stop Band
Values - entries entered into table least to most, also with colon e.g. 9:1234.
Return with no data entry gets you to next page.

<u>Page 6</u> Pause Duration
Select wait period.

<u>Page 7</u> Data Acquisition
Printer
Run ID
Date
Shift + or sequences through selections.
Return gets to next.

Page 8 Instrument Control

Power	Close Shutter	ON
Hi Volt	Test Lamp	

Page 9 Number of Test Runs And Run Control Data

Page 10 Data Summary

Run Control

Keyboard Pause ON	= CTRL-S
OFF	= S
Test No. Clearing	= CTRL-C
Printer OFF	= CTRL-O
Printer ON again	= CTRL-O
Stop Mode, Step Advance, Data Entry	= CTRL-E
Advance to next Stop Value	= S

Data Summation

= 32 bit accumulation

The Program is not smart enough to recognize improper mode control selections and will stall if instrument power is OFF. Reset is necessary to recover.

Stop Mode

Let instrument advance to first stop value and print output data.
Stop printer with CTRL-O.
Enter instrument step advance data with CTRL-E.
Note Cursor at data entry position on Page 10.
Enter step advance desired. Depress RETURN.
Verify that instrument steps desired count.
Restart printer with CTRL-O.
Stop printer with CTRL-O and repeat above for as many data points as necessary.

Pause, Stop, Band data points do have to be sequential because the compare routine will stop the table scan when it finds a value larger that the current STEP value that is being searched for in the table.

IV. Optical Performance Verification

A. Install spectrometer in vacuum system.

B. Attach all cabling, sources.

C. Pump Down system to 1Ø -6, TORR, stabilize.

D. Operate System as described in III.

E. From source data, verify that spectrometer is scanning the required wavelength range (1Ø5Ø A^{0} to 18ØØ A^{0}).

F. From Spectral line shapes, verify that spectral resolution is proper for the slit width (5 A^{0}), thus assuring proper optical alignment.

V. Environmental Testing

The complete spectrometer package will be subjected to thermal vacuum testing, and optical calibration testing. Test parameters and levels for these tests are to be determined at this time. These criteria can be established when more is known about the launch vehicles. The spectrometer will be operated before, during and after each test to assure proper operation.

APPENDIX C

Test Plan No. RSI 79-C-0044-002 Prepared August 6, 1980

ACCEPTANCE TEST PROCEDURE

HORIZON UV 801 EXPERIMENT

PROTOTYPE TELESCOPE

Air Force Geophysics Laboratory
Hanscom Air Force Base
Massachusetts 01731

Research Support Instruments, Inc.
10610 Beaver Dam Road
Cockeysville, Maryland 21030

ACCEPTANCE TEST PROCEDURE
FOR VERIFICATION OF FOCUS
AND ALIGNMENT OF PROTOTYPE
TELESCOPE
HORIZON 801 UV EXPERIMENT

I. Equipment Required

A. Prototype Telescope, Model 140-193-0-1, S/N 001

B. Optical Bench and Accessories

C. Red Laser

D. Source Lamp

E. Alignment Fixture

1. 140-193-0-21, Centering Target, Sun Shade End
2. 140-193-0-22/1, Centering Target, Mirror End
3. 140-193-0-22/1, Focal Plant Target

F. Flat Mirror

II. Preliminary Preparation

A. Laser Alignment

1. The laser must be mounted on the optical bench, and aligned so that its beam is parallel with the bench ways. This is accomplished by mounting the laser on the bench, and mounting a pin hole target directly in front of the laser. Adjust the target to pass the beam, then move the target to the far end of the bench still in the same relationship to the bench ways, and observe that the beam passes through the target. If not, adjust the laser mounting. Repeat process until the beam passes through the target in both positions. The laser bean will then be parallel to the bench ways.

B. Telescope Mounting

1. From telescope assembly, 140-193-0-1, remove the sun shade, mirror cell and mirror from main housing. Mount the housing on the end of the optical bench opposite the laser. Attach fixture 140-193-0-21 to front end of telescope and fixture 140-193-0-21/1 to mirror end of telescope. Observe that the laser beam passes through both targets. If not, adjust the telescope housing on the bench until the beam

passes through both targets assuring that the longitudinal axis of the telescope housing is parallel with the laser beam, and that the beam is entering the telescope on the central optical axis. Clamp telescope firmly in this position.

III. Telescope Alignment

A. Mirror Alignment

1. The target (140-193-0-21/1) at the mirror end of the telescope must be removed, and the mirror and mirror cell must be installed. After installation, turn on laser and verify that the laser beam passes through the upper hole in target 140-193-0-21. If it does not, adjust the telescope mirror by moving the mounting screws on the mirror cell until the beam passes through the target hole. This adjustment will place the mirror in correct alignment in the telescope housing.

B. Mirror Focus

1. Install target 140-193-0-22/1 at the exit area of the telescope housing. This target centers a pin hole exactly in the focal plane of the telescope, coplanar with the entrance slit of the spectrometer, which will ultimately be attached. Remove the front target (140-193-0-21). Mount an incandescent light source outside the pin hole, so that it shines through the pin hole and illuminates the parabolic mirror. This light is then made parallel by the mirror and reflected backwards out through the telescope. A flat mirror is placed on the optical bench in front of the telescope. This mirror autocollimates the light and reflects it back into the telescope, to be refocussed on the entrance pin hole. The alignment of the flat is verified by using the laser on the backside of the mirrored surface, and reflecting the beam back on itself.

2. The reversing beam should be focussed in the entrance pin hole. If the beam is slightly off the pin hole, fine adjustment of the mirror screws will bring it in. By removing the focal plane target and reversing the mounting feet, the target can be moved by hand in and out of the focal plant. By carefully doing this the spot size can be visually observed for change. If out of focus, the mirror position can be shifted by removing pins, then replacing the mirror, and readjusting if necessary.

IV. Telescope-Spectrometer Combination Checkout

A. Remove test targets from telescope housing. Install Spectrometer (17-193-Ø-1) onto telescope. Remove locking screws in grating arm and move grating to zero order (see details in Test Plan RSI 79-C-ØØ77-ØØ1). Turn on red laser and observe that the beam passes through the telescope, through the center of the spectrometer entrance slit, through the spectrometer optical system, and out the exit slit (detector removed). This verifies that the telescope and spectrometer are aligned, and will function as one system.

V. Telescope Field-Of-View Verification

A. To check the .Ø5 degree total field of view of the telescope the system, (telescope, spectrometer, GSE) should be set up completely with the spectrometer/telescope mounted in a vacuum chamber, fitted with a point source and a means to accurately move the spectrometer in a rotating fashion, pivoting about its entrance slit from side to side. Positioning and readout must be extremely accurate, since the total field of view is only .Ø5 degrees, with the half angle .Ø25 degrees.

B. Line up the spectrometer with the source to that the source is exactly on the central ray of the optical system. After pumpdown, position the spectrometer drive so that the grating is sitting on a spectral line peak. Observe that output signal on the GSE. Carefully move the spectrometer, telescope assembly from side to side. Observe the signal drop off and note the angle. This should be .Ø25 degrees on either side of center.

VI. Environmental Testing.

The telescope assembly will be subjected to thermal vacuum and shock and vibration testing, with levels to be determined at this time. These tests will be performed with the telescope mounted on the spectrometer. Alignment verification tests will be performed before and after each test.

APPENDIX D

REPORT NO. RSI-Ø33 DATE 24 JUNE 1985

EQUIPMENT INFORMATION REPORT

"SACS" EXPERIMENT

TWO SPECTROMETER PACKAGE

WITH SCAN PLATFORM

F-19628-79-C-ØØ44

PREPARED FOR

AIR FORCE GEOPHYSICS LABORATORY

HANSCOM AIR FORCE BASE

MASSACHUSETTS Ø1731

PREPARED BY

RESEARCH SUPPORT INSTRUMENTS, INC.

1Ø61Ø BEAVER DAM ROAD

COCKEYSVILLE, MARYLAND 21Ø3Ø-2288

TABLE OF CONTENTS

AUTHORS: Bob Eppig
Tom Riley
File: SACS
DISK: Job 612

"SACS" EQUIPMENT INFORMATION REPORT

1.Ø GENERAL

The purpose of the "SACS Experiment Package" is to make spectral measurements of three areas. One, the "shuttle glow" by looking at the tail of the spacecraft. Two, the plume of a burning engine by looking at one of the orbital maneuvering engines. Three, a reference or dark count by looking into deep space.

The experiment consists of two spectrometers with attached telescopes and detectors, one of which scans an area of the ultraviolet spectrum and the other scans in the visible range of the spectrum. Accessories included with the spectrometers and telescopes are motorized dust covers for each telescope, dark shutters for each spectrometer, mercury test lamps, solar sensors, and a nitrogen purge system for all optics. The spectrometers and telescopes are mounted on a platform which also provides attachment points for the whole package. The platform is motorized so it can position the spectrometers and telescopes to look at the three measurement areas.

The experiment has its own electronics which interfaces with spacecraft telemetry. The electronics carries out commands from the spacecraft, processes data from the spectrometer and the accessories, formats the data and sends it back to the spacecraft.

2.0 SPECTROMETERS GENERAL

The basic spectrometer is an Ebert Fastie 1/8 meter focal length system with an effective aperture ratio of f/5. Theoretical wavelength resolution is better than 1 A. In an Ebert spectrometer light comes in the entrance slit and is collomated and reflected off one half of the Ebert mirror to the grating. The grating reflects the diffracted rays back to the other half of the mirror which focuses the light out the exit slit. To look at different wavelengths the grating is rotated so that the spectrum moves across the exit slit.

2.1.1 OPTICAL SYSTEM; ULTRAVIOLET SPECTROMETER

This spectrometer is configured with a 3600 line/mm grating blazed at 2400A and straight slits, with the entrance and exit slits 1mm wide to give an optical resolution of 20 A, and a field of view of .27 degrees when used with the attached telescope. The slit height will be masked to 7mm.

Since this spectrometer will be used in the ultraviolet region of the spectrum, scanning from 1600 A to 2900 A, the grating and the Ebert mirror are overcoated with MgF_2.

2.1.2 OPTICAL SYSTEM, VISIBLE SPECTROMETER

This spectrometer is configured with a 1200 line/mm grating blazed at 5000 A and straight slits, with the entrance and exit slits 360 y wide to give an optical resolution of 20 A, and a field of view of .09 degrees when used with the attached telescope. The slit height will be masked to 7mm.

Since this spectrometer will be used in the visible region of the spectrum, scanning from 5000 A to 8500 A, the grating and the Ebert mirror are overcoated with SiO2.

2.2 MECHANICAL SPECTROMETER GENERAL

The spectrometer is a modified R.S.I. #10-125-0-1 Ebert Spectrometer and has demonstrated itself to be an extremely reliable and rugged instrument. It has been modified by moving the grating drive mechanism off the slit plate to make room for a bigger detector, and mounting it to the body of the spectrometer.

The spectrometer is designed as a unit with three sections. The first, the main housing, is a cast rectangular tube. This serves as a mounting for the grating drive mechanism, and a spacer, accurately machined to separate the slit plate with the grating from the Ebert mirror.

The second section, the mirror cell, attaches to the rear of the main housing. This cell holds the Ebert mirror in position between nonmetallic surfaces, in such a way as to allow adjustment of mirror in the 3 axes for optical alignment. The mirror cell also contains a purge fitting allowing the mirror and grating to be purged with N2 as required.

The third section, the slit plate, attaches to the front of the main housing. This plate serves as the mounting surface for the slits. The grating in its holder is also mounted to this plate. The slit plate of the spectrometer becomes a central structural member of the experiment package, providing mounting points to the scan platform as well as mounting for the telescope, detector, dark shutter, and alignment mirror.

The grating drive mechanism causes the grating to move through its desired scan angles. It is powered to by a 15 degrees stepper motor run through a 24.32 to 1 gearbox which is attached to a cam. The cam rotates 360 degrees in 2.9 seconds if the motor is stepped at 200 steps per second. The rise and fall of the cam causes a slide to move back and forth. An arm which is attached to the mounting box of the grating, comes through the side of the main housing and rides in a slot in the slide. Therefore, as the slide moves it imparts a sinusoidal motion in the grating. Also, there is a fiducial reader associated with the mechanism which reads a spot on the cam once every revolution as a reference point for each scan.

The spectrometer is black anodized and baffled to reduce reflections and scatterings of light. The slide, as well as all optical components, has adjustments for optical alignment and for setting the wavelength. All this is shown on Drawing No. 15-224-0-1.

2.3 ALIGNMENT MIRROR

An alignment mirror, which can be attached to the slit plate of the spectrometer, is mounted so that a normal to its surface is parallel to the entrance optical axis of the spectrometer. It must be removed and attached to the telescope when the telescope is mated with the spectrometer.

2.4 MATERIALS

The basic contruction of the spectrometer is aluminum, with Vespel used for the cam and the slide. Vespel is used because it has good sliding properties with low wear. All the optical components are made from precision annealed cervit mounted against kel-F plastic. All other parts and hardware are non-magnetic stainless steel. In order to assure good vacuum operation, all bearings in the instrument, gearbox, and gears in the gearbox are lubricated with a vacuum compatible lubricant from Bray Oil Company, called Braycote #601 and / or BrayCo #815Z. All screw fasteners equal to or bigger than 2-56 are locked into place with locking heli-coils. Smaller than 2-56 are locked in place with an approved epoxy.

3.0 TELESCOPE GENERAL

The telescope unit is a mechanical-optical assembly which, when attached to the spectrometer, allows light to be collected and focused in the spectrometer. The telescope is designed to match the optical throughput of the spectrometer, and to be consistent with the requirements of the complete spacecraft instrument system. Two telescopes will be used; one on the ultraviolet spectrometer, and the other on the visible spectrometer.

3.1 OPTICAL DESIGN

The principle criteria for the optical design are that the telescopes be f/5 to match the spectrometers, and the physical contraints of small size weight. The following equation was used to determine the optical design.

$D/F = 1/f$ where : D = Diameter of collector

F = focal length

f = f number

With this equation and the principle criteria in mind, the following design evolved:

D=Diameter of mirror =46mm

F=Focal Length=230mm.

The mirror diameter must, in actual practice, be slightly larger to accommodate the slit height of 7mm.

3.2 OPTICS

The single optical element in the telescope is a parabolic mirror, due to its better image properties for off axis light. In this case the off axis light is at a very small angle, but the parabola will still give better results than a spherical mirror. This mirror is made on a blank of 55mm x 68mm x 9mm thick at its center. The blank is made from precision annealed cervit, and weight relieved to keep the mass small. The front surface is precision ground and polished to an off-axis parabola with a 230mm focus. The mirror is mounted in the telescope such that its in coming central ray makes an angle of 12.093 degrees with the central entrance ray of the spectrometer.

3.2.1 OPTICS, ULTRAVIOLET TELESCOPE

Since this telescope is to be attached to the ultraviolet spectrometer and used in the region from 1600A to 2900A, the parabolic mirror will be overcoated with MgF_2.

3.2.2 OPTICS, VISIBLE TELESCOPE

Since this telescope is to be attached to the visible spectrometer, and used in the region from 5000A to 8500A, the parabolic mirror will be overcoated with SiO_2.

3.3 ALIGNMENT MIRROR

A small alignment mirror will be provided and mounted such that its surface is normal to the central entrance ray of the telescope. This mirror will be mounted under the telescope aperture, between the telescope housing and the spectrometer housing. This is shown in drawing #140-193-0-1 A.

The alignment mirror is the same mirror used as the spectrometer alignment mirror but relocated when the telescope is assembled to the spectrometer.

3.4 MECHANICAL DESIGN

The main telescope housing is designed as a riveted structure. This type of structure has been used successfully in the past for baffles and telescopes. The telescope assembly is designed as a unit with three main sections. The first, the main housing, is a long rectangular tube, made as four plates and riveted together to form a rectangular baffle tube. The plates are internally finned to provide multiaperture baffles along the entire length. The edge joints are designed with a double step, providing an effective light trap when completely assembled. The flat wall which forms the section of the telescope that covers the spectrometer drive motor cannot be fitted with large baffle surfaces, since the light path is extremely close to this wall. However, the surface has been scored, forming a saw-tooth effect on the surface, with the depth equal to the separation. This should provide additional baffling against grazing rays, and not intrude on the field of view. Details of the above description are shown on drawing #140-193-0-1 A. This main section of the telescope assembly also contains a mounting surface for a solar sensor and an Hg test lamp with provisions in the baffles to allow the Hg lamp to illuminate the parabolic mirror. The main section has included in it a motorized dust cover, which when closed, blocks off the baffle tube and is used to keep contaminates off of the telescope mirror. The cover remains closed until measurements are ready to be taken.

The second section, the mirror cell, is an aluminum holder which attaches to the rear of the main housing. This cell holds the telescope mirror in position between nonmetallic surfaces. The mirror cell also contains a purge fitting allowing the mirror to be purged with N2 as required.

The third section, the sun shade, fits on the front of the main housing. This section is machined from aluminum and is designed to keep the sun's rays from entering the field of view of the telescope. The entire assembly is attached to the spectrometer by mounting flanges which mate to the front plate of the spectrometer.

The entire telescope assembly will be black anodized to reduce scattered light. The complete mechanical assembly is shown in drawing

#140-193-0-1A. Telescope assembly.

3.5 MOTORIZED DUST COVER

The dust cover is mounted to the main housing of the telescope. The cover is a sheet metal slide, which rides in a support housing, and has two positions, open and closed. A 120 pitch rack, which is attached to the slide, mates with a 30 tooth gear mounted on a 24.32 to 1 ratio, size 8 gearhead. The motor used is a 15 degree size 8 stepper motor. When run at 200 motor pulses per second the cover will open or close in 7.6 seconds. There are two fiducial readers used. One senses the open position, and the other the closed position.

3.6 MATERIALS, TELESCOPE AND DUST COVER

The basic construction of the telescope and dust cover is aluminum, with 660 bronze used as supports for the rack in the dust cover. The optics are made from precision annealed cervit mounted against kel-F plastic. Teflon is used on the sliding surfaces of the dust cover. All other parts and hardware are non-magnetic stainless steel. In order to assure good vacuum operation, bearings and gears in the gearbox are lubricated with a vacuum compatible lubricant from Bray Oil Company, called Braycote #601 or BrayCo #815Z. All screw fasteners equal to or bigger than 2-56 are locked into place with locking heli-coils. Smaller than 2-56 are locked in placed with an approved epoxy.

4.0 SCAN PLATFORM GENERAL

The scan platform is an electro-mechanical system which can mount two Ebert spectrometers with telescopes, and allows them to scan through a total angle of 75 degrees (in its configuration as shown in drawing #0-224-0-1 "E"). This system, when correctly oriented on a Space Shuttle, allows the spectrometers to be pointed at one of the "orbital maneuvering engines" and then scanned up through the Shuttle tail section and into deep space. The scanner is driven by a stepper motor and controlled by its own electronics package which allows positional control and readout.

4.1 SCAN PLATFORM, MECHANICAL

The scan platform is designed as a central compact structure supporting a trunnion assembly which mounts the spectrometer and telescope assemblies. The scan mechanism employed is a stepper motor driven worm, turning a worm wheel, which in turn moves the spectrometers and telescopes which are attached to the worm wheel by a bearing mounted trunnion mechanism. Two gears are used between the stepper motor and worm to allow the locating of the worm and worm wheel and stepper motor in the smallest possible area. This configuration requires the stepper motor to be reversed in order to return the instruments to the starting position. The worm is driven by a size 18, 15 degree stepper motor through a 17 to 1 gearbox and a 28 tooth and 20 tooth geartrain. One revolution of the worm produces 3.75 degrees of scan. It takes 291.43 steps to produce one revolution of the worm. Since the total angle of scan is 75 degrees, it will require 20 revolutions of the worm or 5828.6 steps. Therefore, it will take approximately 30 seconds to scan 75 degrees when the motor is run at 200 motor steps per second. Since the field of view of the instruments is .27 degrees for one, and .09 degrees for the other, and since one motor step provides .013 degrees of scan, the system gives adequate angular resolution for good positional data.

The scan platform also includes a locking mechanism to lock the instruments in one position during the lift-off and landing phases of the Shuttle mission. This prevents unnecessary vibration and movement during these times of high vibrational and shock loads. The lock is a steel rack which is driven into a slot preventing movement. The rack is driven by a 28 tooth gear mounted directly on the shaft of a D.C. permanent magnet planetary gear motor. The locking rack moves approximately 1/2 inch or approximately 8 teeth and the 28 tooth gear which drives the rack moves 103 degrees or .286 revolution. The motor turns the gear at .148 revolutions per second, therefore, it locks or unlocks in approximately 2 seconds.

4.2 MATERIALS

The basic construction of the scan platform is of aluminum with stainless steel shafts. The gears and worm are also aluminum and stainless steel. The worm wheel is bronze. All moving elements are ball bearing mounted, and preloaded to remove play. All bearings, gears, worm, and worm wheel are lubricated with a vacuum compatible lubricant from Bray Oil Company, called Braycote #601, and or BrayCo. #815Z. All screw fasteners equal to or bigger than 2.56 are locked into place with locking heli-coils. Smaller than 2.56 are locked in place with an approved epoxy.

4.3 INTERFACING

The scan platform is designed with a general mounting bracket, since no specific interface requirements have been established. The logic electronics are in a housing which will mount on the scan platform in position shown on drawing #0-224-0-1 "E" Sheet "2". Detailed drawings will be provided when they become available.

NOTE: If mounting configuration changes (ex. angles changes, or scan angle changes) mounting and operating area of instrument will change.

5.0 MISCELLANEOUS ELEMENTS

5.1 ENTRANCE SHUTTER

The entrance shutter is a mechanical device which when actuated blocks the entrance slit of the spectrometer for dark count readings. It is mounted directly over the entrance of the spectrometer's front plate.

5.1.1 ENTRANCE SHUTTER MECHANICAL

The main item in the entrance shutter is a pancake type D.C. torque motor mounted in a housing. The motor shaft which protrudes from the housing has a vane attached to it. When the motor is energized, the vane moves and covers the entrance slit, blocking the light entering the spectrometer. When the motor is not energized the vane is spring-loaded in the open positon.

5.2 PURGE SYSTEM

The two telescopes and two spectrometers can be purged with dry nitrogen as required. There are purge fittings on each telescope and spectrometer near the optics associated with each item. All the fittings are connected together with an access port (quick disconnect) located in front of the experiment package.

6.0 ELECTRONICS

The electronics of the two SACS instruments are designed so that additional simular instruments may be added on later flights with only minor changes in hardware and software. Each instrument will have a microporcessor card, integrated detector, dust cover, shutter, and wavelength motor box. All instruments will share a power supply box, a spacecraft interface card, a scan platform motor box, and a sun sensor. One of the Instruments will be designed the primary instrument and will control the scan platform.

6.1 MAJOR ELECTRONIC SYSTEMS

The SACS electronics will consist of the following systems.

6.1.1 MICROPROCESSOR CARD

Each instrument will have a mcroprocessor card containing a

8085 microporcessor system, a 16 bit data counter, output ports, input ports, sun presence circuit, interrupt circuits, and three analog outputs. All microporcessor cards will be capable of serving as the primary instrument but only one will have the scan motor cable attached and will have the scan software active.

The input ports will support the following functions:

a. Read the fiducials
b. Read the command buffers
c. Test for the sun presence circuit trip
d. Check the high voltage level detector
e. Check the calibration bit
f. Check the primary instrument bit
g. Read data counter
h. CHeck if next instrument has completed a data cycle
i. Scan Platform motor attached bit.

The output ports will support the following functions:

a. Step the waelength motor
b. Step the scan motor (primary instrument only)
c. Output the data
d. Enable the next instruments to send data
e. Operate the shutter
f. Control power to the detector
g. Control power to the test lamp
h. Override the sun sensor circuit
i. Signal next instrument that data cycle is complete
h. Operate Dust Cover.

The interrupt circuits will enable the microprocessor to stop, then:

a. Read and transmit data
b. Accept a new program. (EEPROM operation only, see 6.2.2)

The three (3) analog output circuits will monitor the detector high voltage, the temperature of the detector electronics, and the temperature of the microprocessor electronics. The detector high voltage will be monitored by a level detector which the microprocessor can read as one bit.

The microprocessor cards are mounted in a single box with the

Spacecraft Interface card. These cards do not require headsunk components so they may be designed as removable cards. All connections to the cards are made with subminiature D or coaxial connectors; no edge connectors will be used. Future additions of microprocessor cards may be accommodated by either having extra slots or by extending the box. The box backplane ends in a series of terminating resistors so the last microprocessor card can tell that it is the last one in line.

6.1.2 INTEGRATED DETECTOR

The integrated detector will consist of a photomultiplier tube, high voltage supply, and a output pad. The output is a TTL compatible signal sent over a coax cable. An analog high voltage monitor and temperature signal are also provided.

6.1.3 DUST COVER

The dust cover is motorized and has independent full open and full closed fiducials.

6.1.4 WAVELENGTH MOTOR BOX

The motor step signals from the microprocessor card do not have enough strength to directly drive the wavelength stepper motor. This box contains heatsunk transistors to drive the motors and a power filter. Similar transistor drives for the dust cover and shutter are also included in this box.

6.1.5 POWER SUPPLY BOX

This unit will provide filtering and power regulation to all other systems. Heatsunk regulators provide the +5 voltage, and separate filtering is provided for electronic and motor power.

6.1.6 SPACECRAFT INTERFACE CARD

This unit will buffer the command signals from the spacecraft and buffer the data transmission to the spacecraft. The command signals are optically isolated and protected from high voltage. The data enable and clock signals are buffered and sent to the microprocessor cards. Data from the microprocessors is assembled into a single bit stream and buffered. This card does not contain a microprocessor.

Eight (8) command lines are read. These may be interpreted by the software as either 8 independent one bit commands or as a seven bit number and a command valid bit (allowing 125 commands). If the seven bit number is used, four additional data bits will be needed.

This card will be mounted in the same box as the microprocessor cards.

6.1.7 SCAN MOTOR BOX

This box contains the drive transistors for the scan motor and latch motor. It will also contain a power filter. Only the primary instrument requires this box.

6.1.8 SUN SENSOR

This sensor will signal the presence of the sun in the field of view and allows all instruments to start a safety shutdown. One sun sensor operates all instruments.

6.1.9 TEST LAMP

The test lamp provides a source of the Hg spectral lines in each instrument. Its power is controlled by the microprocessor. Each instrument has its own test lamp.

6.1.10 SHUTTER

The shutter on each instrument blocks the light path and is used for dark counts. It has a single fully closed fiducial and is controlled by the microprocessor.

6.2 SOFTWARE

The software watches for a command to be sent from the spacecraft and then initiates a series of instrument responses to the command. At periodic intervals the instruments data is read, organized, and transmitted to the spacecraft.

6.2.1 COORDINATION OF ACTION

The software for the two instruments is almost identical. Both the instruments have the scan platform software but only the one designated as the primary unit runs it. Both units respond to the houskeeping commands such as "Open Dust Cover" in the same way, but their timing may be staggered.

The response of the two instruments to the major experiment commands will be different but coordinated. The exact actions such as "Go To a Specified Wavelength" will be chosen from a standard library. At any given time the instruments may be doing different library functions and be using different constants, but their actions are synchronized at the start of each major data reading cycle. Timeouts will be included in the software so that no instrument waits too long for the other instruments.

6.2.2 LANGUAGE

There are two possible microprocessor languages for this project. The choice between these two is being evaluated at this time.

6.2.2.1 FORTH

Forth is a machine control language originally developed for radio telescopes. When compared to assembly language, it cuts development time by a factor of 5 while keeping most of the microprocessors operating speed. The software rewrite time between flights would also be cut dramatically.

The Forth code would be written on an IBM PC and transferred to EEPROMs (Electrically Erasable Programmable Read Only Memory) in the flight instruments. The use of EEPROMs, which are electrically programmable and erasable, allows the whole process of developing and debugging the software to be done through a single RS-232 connector on the microprocessor box without the need for continually opening the box to change components. This will be particularly advantageous for last minute changes. A second interrupt circuit will allow new programs to be transferred to the EEPROM.

The use of Forth would require more memory than assembly language so a few more chips would be needed. The Forth language core and most of the required support software are available from two suppliers. Some additional support software will be written by RSI.

The use of the EEPROMs will require an RS-232 port requiring one extra chip and a connector. We are currently assembling information on the reliability of available EEPROMs. Some of the advantage of Forth is lost in this application if acceptably reliable EEPROMs are not available.

6.2.2.2 ASSEMBLY LANGUAGE

The program would be written in 8085 assembly language on a IBM PC and transferred to the insturment on UV erasable PROMs (Programable Read Only Memory) or by the EEPROMs and RS-232C port described by 6.2.2.1. The execution time of the finished program is the fastest possible and a minimum chip count is required.

The development and debugging time for this system is long and the exact time required for debugging is difficult to predict.

The simplest implementation of an assembly program would use the UV PROMs, which have a longer reliability history then EEPROMs, and would require the software to be transferred to the instrument by burning the PROMs outside the instrument and physically plugging them in. This process slows developemnt time and is difficult to do in the field, even for very small changes.

EEPROMs and the RS-232C port could be used with assembly language if R.S.I. developed monitor software. This would allow the software to be changed, even at the last minute, from outside the instrument without opening it. The need to use the microprocessor emulator (not used with Forth) would require access to the circuit board for major program development and debugging. As mentioned above, this approach does require a few additional chips.

6.2.3 SOFTWARE STRUCTURE

The software will be written in a structured form. The task will be broken down into component blocks, each with only one input and one output, which are called in proper sequence by a main program.

6.3 DATA STRUCTURE

The data stream is structured so that the primary instrument provides data about the scan platform and its spectrometer, then the secondary instrument adds data from its spectrometer. The number of data bits from each instrument must be divisible by 8. The bit assignments are as follows:

a. Scan Platform (primary instrument only)

Command	3	
Angle	13	
Angle Fiducial	1	
Latch Fiducial	1	
Sun Presence	1	
Subtotal (a.)		19

b. Data Primary Instrument

Count	13	
Wavelength	10	
Wavelength Fid.	1	
Cover Fid.	1	
Shutter Fid.	1	
High Voltage	1	
Test Lamp	1	
Error	1	
Spare Bits	0	
Subtotal (a. + b.)		48

c. Secondary Instrument

Count	13
Wavelength	10
Wavelength Fid.	1
Cover Fid.	1

Shutter Fid.	1		
High Voltage	1		
Test Lamp	1		
Error	1		
Spare Bits	3		
Subtotal (c.)		32	
Total	80		Bits

The time interval between data samples (set at .01 sec in the previous instrument) will be limited by the allowed data transmission rate. The use of a microprocessor in each instrument allows each instrument to be interrupted only during its own data transmission cycle. This means that each instrument is capable of taking and transmitting data as fast as the original single instrument.

6.4 FUTURE ADDITIONS

The system can accommodate future additions of instruments. These instruments must be of the same general design but may be set for different wavelength (different PIM's, and gratings). The following changes will be needed for instrument additions:

Hardware
- a. Additional instruments
- b. New mounting base
- c. Additional yoke supports
- d. New cables
- e. Additional microprocessor cards
- f. New or modified spacecraft interface card
- g. New or extended card box

Software
- a. Normal interflight command changes
- b. Data transmission adjustments (if necessary)

The only limit on the number of instruments is the data transmission rate, each instrument requiring 32 additional bits. The additional instruments may be mounted on the same scan platform, fixed to the spacecraft, or mounted on a separate platform.

APPENDIX E

ELECTRONICS MANUAL

FLIGHT INSTRUMENT

SACS

JOB 612

Contract No. F19628-79-C-0044

Research Support Instruments, Inc.
10610 Beaver Dam Road
Cockeysville, Maryland 21030-2288
(301) 785-6250

J. T. Riley
06/17/86
DISK: SACS 612

Files: FIM612
XSOFT612
XMAP612
APPEN612
FP.SCR

TABLE OF CONTENTS

FIGURES

ASSOCIATED PRINTS

Microprocessor Card Schematic	8ØØ-224-4Ø-4
Motor Driver Card Schematic	8ØØ-224-Ø-8

CONTRACT CONTACT

Charles Forsberg - LIU
AFGL
Air Force Systems Command
USAF
Hanscom Air Force Base
Mass. Ø1731

(617) 337-2625

FLIGHT INSTRUMENT ELECTRONICS

GENERAL DESCRIPTION

INTRODUCTION

This document is the primary written reference for the electronics of the SACS 612 project. It serves as the design specification and the operating manual for both the flight instrument and the Software Development System. The hardware and software for the system are described in detail.

1.0 Purpose of Document

This document serves as the specification for the electronics portion of this project. The function of the electronic hardware and software are described in detail.

Also included are instructions for the operation of the Software Development System. These provide step by step instruction for the writing of software, its debugging, and the burning of PROMs.

The optics and machinical design are not covered.

1.1 Description of Flight Instrument Electronics

The flight electronics is composed of a number of printed circuit cards mounted in individual aluminum boxes and connected by cabling. This system controls the flight instrument in response to commands from the sapcecraft and transmits data to the spacecraft.

1.2 Description of Software Development System

The Software Development System consists of a small computer, power supply, and cables which support development of the flight software and debugging of both the software and hardware.

SACS HARDWARE

INTRODUCTION

The hardware for the SACS, a two spectrometer instrument with scanning platform, is described in detail. This same electronic system may be used with two or four spectrometers and either with or without a scanning platform. The hardware for the Software Development is also described.

2.0 Hardware Description

The electronics for the SACS multiple spectrometer instrument is divided into the following modules:

1. EMI Filter (1 per system)
2. +5 Power Supply (1 per system)
3. Spacecraft Communications Module (1 per system)
4. Microprocessor Module (1 per spectrometer)
5. Motor Control Module (1 per two motors)
6. Integrated Detector (1 per spectrometer)

Also on the instrument are the following electronic devices:

1. Wavelength Motor (1 per spectrometer)
2. Wavelength Cam Fiducials (1 per spectrometer)
3. Dust Cover Motor (1 per spectrometer)
4. Dust Cover Fiducials (2 per spectrometer)
5. Shutter (1 per spectrometer)
6. Test Lamp (1 per spectrometer)
7. Platform Motor (1 if platform used)
8. Platform Cam Fiducials (2 if platform used)
9. Platform Latch Motor (1 if platform used)
10 Platform Latch Fiducials (2 if platform used)
11 Sun Sensor (1 or 1 per spectrometer)
12 Flight Cables (1 set)

The system (see Figure 21) consists of the basic support equipment (EMI filter, Power Supply, Spacecraft Communication Module, and cables); a Master Microprocessor (Microprocessor Module, Motor Control Module, Integrated Detector, motor, etc.) and one or three Secondary Modules (identical hardware with Master). If the system has a scan platform then an additional Motor Control Module will be cabled to the Master Microprocessor.

Each module will consist of a single printed circuit board mounted in a machined aluminum housing with subminiature D connectors. The components are chosen for highest reliability. All components which dissipate significant heat will be heatsunk to the module cases.

2.1 Electronic Subassemblies

The following electronic subassemblies are used in the system.

2.1.1 EMI Module

The EMI module prevents electromagnetic from getting from the instrument back to the spacecraft over the power lines. It also smooths and filters the incoming +28 volt power for the instrument.

2.1.2 +5 Power Supply

The +5 power supply converts +28 volt power into regulated +5 voltage for use by the Microprocessor Modules, the Spacecraft Interface Module, the Motor Control Modules, and the Integrated Detectors. This is a standard high reliable flight power supply and its size is dependent on the number of spectrometers in the system.

2.1.3 Spacecraft Communication Module

The Spacecraft Communication Module acts as the interface between all the Microprocessor Modules and the spacecraft. It performs the following functions:

.1 Buffers 8 incoming Command Lines.
.2 Buffers incoming Data Enable and Data Clock lines.
.3 OR's serial data lines from all microprocessors.
.4 Buffers outgoing data.
.5 Provides +10 volt power for analog Circuits.
To a Maximum of 75 milliamps per microprocessor.
.6 AND's Scan Complete signal from multiple
secondary Microporcessors.

2.1.4 Microprocessor Module

The Microprocessor Module is the brain of the instrument and contains the following sections.

2.1.4.1 Forth Core

This is a 8085 microprocessor with a Forth operating system in ROM. It allows the flight software to be developed quickly in a higher level language and supports software debugging. A detailed allocation of the memory is included in Appendix I, Memory and I/O Map.

All the ROM, both that is used for the Forth Core and used for the Flight Program, is of the most reliable fusible type. A Software development RAM chip is included to allow the software to be developed and debugged without burning ROMs. The developemnt is removed before flight to save power and improve reliability.

The core has a RS-232 port which is used only for software development and debugging. The RS-232 driver chip is moved off the flight board and placed in a small separate box associated with the development cable.

2.1.4.2 Data In and Out

The data transmission system is initiated by the Data Enable signal tripping the 7.5 interrupt. The data is then transferred from memory to a 8 bit shift register which shifts the data out on the Data Clock pulses. The Data Clock pulses are also counted in hardware with a divide by 8 so that the timing of the transfer on the next data word is easily accomplished.

2.1.4.3 Fiducials

All the fiducials consist of LED/phototransistor pairs mounted so that a hole in a mechanical device is aligned with them at the reference mark.

The fiducials on the motors read a hole in a disk attached to the drive cam. This produces one reference mark per complete cycle and this mark is located in the middle of the flyback segment.

The dust cover and latch have fiducials at the fully open and the fully closed position.

2.1.4.4 Analog Signals

Up to four analog signals may be input, buffered, and output by the microcomputer card. The outputs are not inverted and have a 2K series resistor. Each of these levels may be compared to a reference level set by a resistor and these comparisons can be read by the microprocessor. If all four comparisons are not needed, a signal may be compared against two levels.

The analog chips may be powered by an external power source on the Spacecraft Interface card or from the internal +5.

2.1.4.5 Scan Complete

To coordinate the wavelength scans of more than one microprocessor, a SCAN COMPLETE INPUT and a SCAN COMPLETE OUTPUT signal are provided. The secondary units set the SCAN COMPLETE OUTPUT high when they complete a wavelength scan. They then wait for the SCAN COMPLETE INPUT to go high or a timeout to occur before starting the next scan. The SCAN COMPLETE OUTPUT is reset on the start of the new scan.

The Primary microprocessor, upon completing its wavelength scan, waits for the SCAN COMPLETE INPUT to go high or for a timeout to occur before starting its next scan and pulsing the SCAN COMPLETE OUTPUT high for 1 millisecond.

For an instrument with more than one secondary microprocessor, the SCAN COMPLETE OUTPUT signals from the secondary units are ANDed together on the Spacecraft Interface Card and the result sent to the Primary unit.

2.1.4.6 Motor Control

The microprocessor may control up to 4 stepper motors but cannot provide enough current to drive the motors. Relay drivers are also provided to switch the motors from a step to a hold state.

2.1.4.7 NOT SUN PRESENT

The NOT SUN PRESENT signal triggers a hardware function to protect the detector from excessive sunlight. The microprocessor can sense the state of this circuit and override the safety if commanded to do so.

When the NOT SUN PRESENT signal is high, a safe and normal condition exists. The Detector Power Relay is energized so that the detector is operational. The Shutter Relay is not energized therefore the Shutter is open.

When the NOT SUN PRESENT signal goes low, the Detector Power Relay is de-energized so the detector high voltage is off, and the Shutter Relay is energized and the Shutter closes. A one minute timer is also started which holds this situation for at least 1 minute. After one minute, if the NOT SUN PRESENT signal has returned to high, normal operation is resumed.

If the microprocessor receives a command 4. HV.OVR. it overrides the effects of the NOT SUN PRESENT signal and the instrument is placed in the normal operating mode. This command is used if the Solar Sensor is malfunctioning. The NOT SUN PRESENT signal is reactivated after a Command 6, Standard Start Position, is received.

If the microprocessor card is powered but the microprocessor is not running, the NOT SUN PRESENT signal going low will lock the detector power off and it will not reset after 1 minute.

The microprocessor may also separately energize the Shutter.

2.1.4.8 Spacecraft Commands

The microprocessor can read 8 command lines from the spacecraft. These are the same 8 lines for all microprocessors but different instruments may respond differently.

2.1.5 Motor Control Module

The Motor Control Module provides drive power for two four phase stepper motors. The motor drivers have input buffers, pulse counter, optical isolators, and headsunk power transistor. Latching relays are included to switch the motors between a holding current and a stepping current.

A Motor Control Module may power either the Wavelength Scan Motor and the Dust Cover Motor or the Platform Scan Motor and the Platform Latch Motor. If a Module is used for the Platform then some of its features will not be used.

Also included in the Motor Control Box are two relays to control power to the Detector and the Cal Lamp. These relays are not latching.

A Shutter driver circuit with optical isolator is also present.

2.1.6 Integrated Detector

The Integrated Detector puts out a pulse for each event detected by the Photomultiplier Tube. The pulses are counted on the microprocessor card.

2.1.7 Motors

All motors are four phase stepper motors.

2.1.8 Fiducials

All fiducials are LED/Phototransistor pairs that put out a low signal when the light path is not interrupted.

2.1.9 Shutter

When energized, the Shutter blocks the entrance slit allowing the reading of dark counts. The Shutter blocks light from the Test Lamp and is spring loaded open.

2.1.10 Sun Sensor

THe Sun Sensor uses a photo transistor to sense the presence of the sun in or near the field of view of the instrument. Its output is a digital signal, NOT SUN PRESENT, which is 0 when the sun is present and 1 when the field of view is safe. This signal is used to protect the detector from excessive sunlight.

The sensitivity of the unit is set by an internal resistor which must be chosen before encapsulation. The field of view of the sensor is set by the mechanical housing.

2.2 Software Support System

To develop the software and make changes between flights, a non-flight Software Development System is used. This system consists of the following components:

.1 IBM Compatible PC
.2 Multipurpose card with RS-232 port
.3 Power Supply
.4 Prom Burner
.5 Printer (optional)
.6 Dvelopment Cables

This system is used for the following functions:

.1 Write flight software.
.2 Debug flight software.
.3 Exercise flight hardware.
.4 Change flight software between flights.
.5 Development Documentation.

It will perform the followint activities.

.1 Write software
.2 Write documents
.3 Store files on disk
.4 Transmit software to flight microprocessor
.5 Serve as flight microprocessor terminal during debugging
.6 Receive the final flight software from the
flight microprocessor
.7 Burn the flight PROMs

One Microprocessor Module is worked on at a time. The basic flight software is written on the Software Development System and transmitted over the RS-232 cable to a Microprocessor Module. There the program goes into temporary RAM. The software can then be tested one section at a time using the Development computer as a terminal. All flight functions except data transmission may be proven on this system without the use of Ground Support Eauipment (GSE).

2.2.1 Software Development Computer

The Software Development Computer is an IBM compatible portable with a multipurpose card having a RSI-232 and printer port, real time clock, and 360K memory. See Figure 20, Software Development System.

A printer is helpful in software and document development but not mandatory.

2.2.2 Power Supply

The power supply provides enough +28 volt power to run bench tests for the entire instrument and small amount of +5 for the RS-232 communication line.

Also mounted on the power supply housing are a Reset button and a Develop/Run switch used in software development.

2.2.3 PROM Burner

The PROM burner can be any available unit which can burn the type of ROMs used in the flight microprocessor cards and talk to the computer. It is only needed at the end of software development, after calibration, and after between flight software editing. It may either have its own card in the computer or communicate through the RS-232 port. Slight adjustments in software may be needed to suit a particular burner.

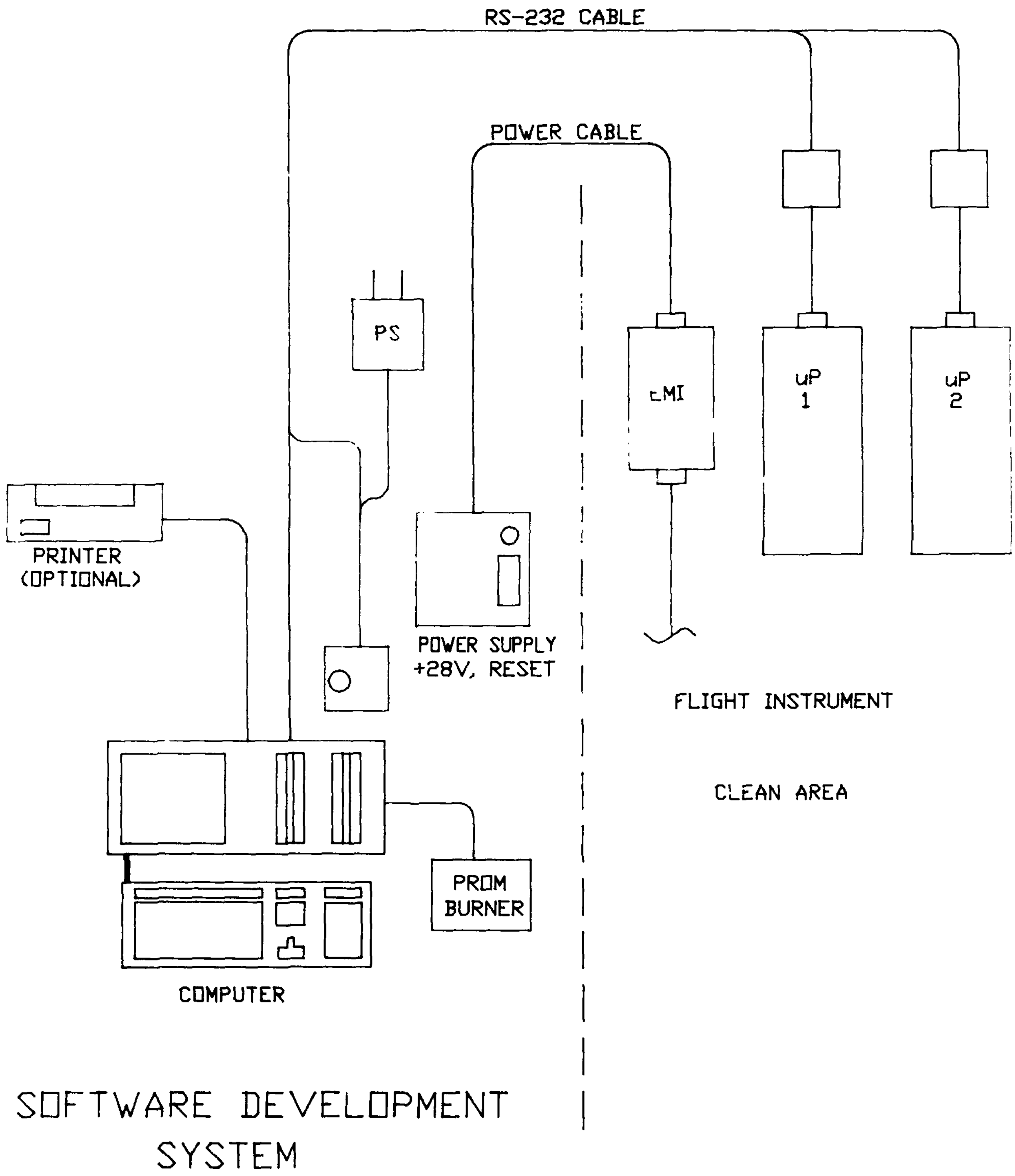
2.2.4 Cables

The cables connect the computer, power supply, and flight instrument. The main cable may be up to 50 ft. long if the flight insturment is to be maintained in a separate clean area.

The flight instrument end of the RS-232 cable terminates in a small box containing a +-12 Volt to +5 Volt signal level converter so that these components will not have to be flown.

FIG 20 SACS 2 INSTRUMENT

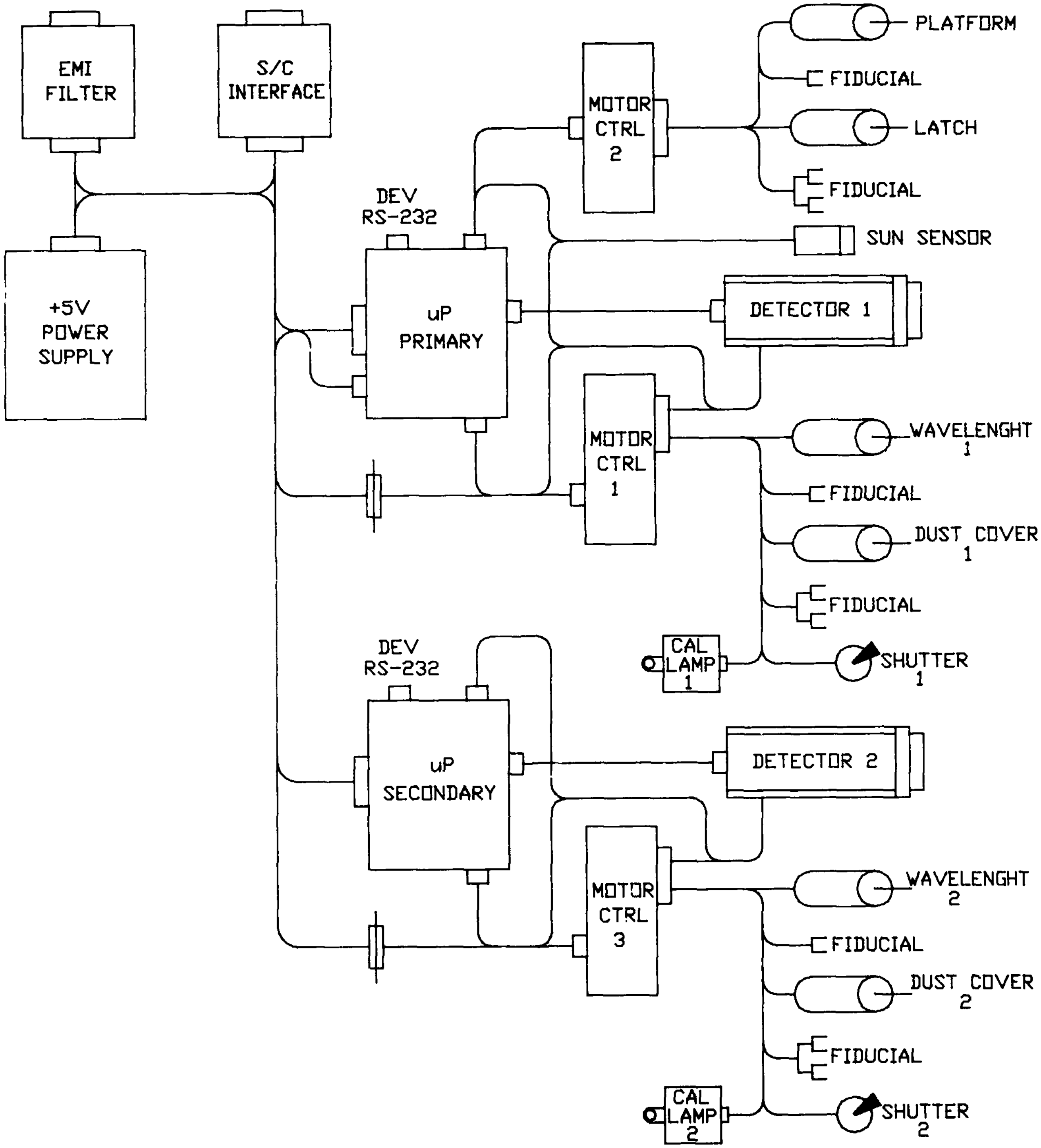
SOFTWARE DEVELOPMENT SYSTEM



RILEY
6-13-86

FIG 21

SACS 2 INSTRUMENT
MODULAR INSTRUMENT CONTROLLER



RILEY
6-12-86

SACS FLIGHT SOFTWARE

J. T. Riley
06/16/86
File: XSOFT612
DISK: 612 DOCUMENTS

INTRODUCTION

The software for the SACS microprocessor cards is first generally discussed and then details of the flight application program are given. Procedures for down-loading the program from the development computer and for burning fusible ROMs are also included.

3.0 Software.

3.1 General Philosophy

The flight instrument software is based on a Forth language core with the application software added in its own ROM or in a temporary development RAM.

In the design phase, the application was broken down into functional blocks each with one entrance and one exit. These blocks are written as words in the Forth dictionary. The large blocks are subsequently broken down into smaller blocks and this process continued until only words in the Forth core, or new assembly language words are needed to write the blocks.

3.1.1 Forth Core

The Forth language core is burned into 6K of fusible ROM and provides a basic set of functions for compiling new portions of the program, interpreting the completed program, and doing math and logic. Also included is a small assembler to allow defining of assembly language words which can also be expanded.

The Forth core is based on the FigForth model with the disk operating system removed. During development the program is stored on the disks of a separate microcomputer, therefore, no disk operating system is needed in the flight instrument.

3.1.2 Structure

The Flight program contains only words needed in flight. Other functions, such as ground calibration, can be supported by temporary programs loaded into the development RAM.

The Flight program is divided into three (3) main Words; INIT which sets up the instrument, DATA which reads and transmits data, and SER.COM which receives and issues commands. Each of these words is broken down into more detailed words.

The INIT initiates the systems variables, resets all commands to the defined starting situation, and puts the instruments through the start-up sequence. This is the Word which is executed on power up, or on pressing the reset button through the RS-232 cable. When this word is completed, execution moves to SER.COM.

The DATA word is executed upon the 7.5 interrupt line which is controlled by the Data Enable signal from the PCM. This word will be run once each time a Data Enable occurs after the INIT is complete. The Data is first transmitted to the Spacecraft Interface Card as a long serial word, and then new data is read and assembled in RAM. The first word of the new data is loaded into the data buffer. At the end of this sequence the elapsed time clock is incremented and execution returned to

SER.COM.

The SER.COM word is an infinite loop and may be left only by hardware reset or powering down. The data interrupt does periodically interrupt SER.COM but its activity is not affected by this. SER.COM first looks at the Spacecraft Command Inputs to determine if there is a valid new command. If there is no new command the program simply waits, rechecking the incoming line periodically. If there is a valid new command, the word for that command is called. The specific command words (SCAN1, SCAN2, FCAL, ect.) produce a sequence of actions such as a pattern of steps to the wavelength motor or a command to the Test Lamp. Between other functions the microprocessor checks for new incoming commands, if a new valid incoming command is found, execution of the present command is stopped, STAD.ST is run, and the new mode sequence started.

3.1.3 Development

The application portion of the software is burned into 4K of fusible ROM. To facilitate development and interflight reprogramming a socket for 4K of development RAM is also provided which can be substituted for the application ROM by means of a special switch on the RS-232 cable. This RAM must be removed before flight as it is not powered during fight and will load down the address lines.

In development, the program is written on the external computer and down-loaded into the RAM through the RS-232 port. After all necessary tests and debugging have been completed, the final flight program is transmitted up the RS-232 cable, stored on disk, and retransmitted to the PROM burner. The Flight PROM may then be inserted in the electronics module and the temporary RAM removed.

A number of utility words have been added to the Forth core to facilitate the development process. These down-load and up-load features assist in burning PROMs.

3.1.4 Ground Testing

The flight instrument may be exercised on the ground through the RS-232 port if desired. This allows most of the instrument functions to be tested without the use of the ground support equipment.

Data transfer may not be fully tested without an additional GSE card for the development computer and additional GSE software.

3.1.5 Flight

In flight the software will:

Initiate the System
- .1 Set up the Software.
- .2 Test for Flight or Development mode.
- .3 Initiate Data reading and transmission.

Handle Data (under interrupt)
- .1 Transmit Data to the Spacecraft Interface Card.
- .2 Read Detector Event Counter.
- .3 Read Status information.
- .4 Assemble the next Data Word.
- .5 Increment the elapsed time clock.

Command the Instruments
- .1 Receive commands from Spacecraft Interface.

.2 Issue steps to the wavelength motor.
.3 Issue steps to the platform scan motor.
.4 Operate the Test Lamps.
.5 Respond to Sun Presence.
.6 Operate Shutter.
.7 Operate Dust Cover.
.8 Read all Fiducials.
.9 Operate platform latch.

3.2 Input

The microprocessor has available the following inputs:

.1 Spacecraft Commands (8)
.2 Fiducial Readings (8)
.3 Data Enable and Data Clock
.4 Detector Counter (16 Bit)
.5 Analog Signal Levels (4)
.6 Not Sun Present (1)
.7 Cal Lamp On (1)
.8 Scan Complete In (1)
.9 Development RS-232 port.

The eight (8) Spacecraft Commands allow the microprocessor to respond to commands from the spacecraft. The software must respond to each of these with a sequence of actions.

The Fiducials give reference marks for moving parts. The eight (8) fiducials for the primary system are:

.1 Wavelength Cam Course
.2 Spare
.3 Dust Cover Open
.4 Dust Cover Closed
.5 Scan Platform Cam Course
.6 Spare
.7 Platform Latch Open
.8 Platform Latch Closed

On the secondary system the Scan Platform and Platform Latch fiducials are also spare.

The microprocessor card senses the Data Enable line and responds with data clocked by the Data Clock. The microprocessor keeps count of the data interrupts and generates an elapsed time to a maximum of 65536 data periods (if data period = .01 sec. then 10.92 minutes). This time information may be used to start and stop actions during the command sequences if the data system is running.

The microprocessor may stop, read, and reset a 16 bit hardware counter for the Detector Pulses.

Three (3) analog signals are buffered in this box and sent to the Spacecraft Interface Card. One of these, the High Voltage monitor is read by a digital level sensing circuit so that the state of the high voltage to the detector may be read.

The Not Sun Present Signal indicates if the instrument is pointed at the sun. Emergency action is taken by hardware and the microprocessor sets the Error Bit.

The Cal Lamp On Signal indicates that the calibration lamp is on.

The Scan Complete Input indicates that the other microprocessor card, has completed a major program action.

The Development RS-232 port is used only for development and debugging software and is not normally used in flight.

3.3 Output

The microprocessor has the following outputs:

.1 Serial Data
.2 Issue Motor Steps
.3 Set Motor Power Level, Step/Hold
.4 Operate Shutter
.5 Operate Cal Lamp
.6 Scan Complete Out
.7 Not Sun Present Override/Reset

3.3.1 Data

The spacecraft sends a Data Enable Signal and a Data Clock to the Spacecraft Interface card. This card buffers these signals and sends them to all Microprocessor Cards. It ORs the returning data signals together, buffers the resulting signal, and sends it to the spacecraft.

The Data Enable line from the Spacecraft Interface card initiates the data transfer by firing the 7.5 interrupt of the microprocessor.

If the unit is the Primary microprocessor card then it immediately issues the latest data word which it has previously stored in its memory. The data is clocked out by the Data Clock. If the microprocessor card is a secondary card, it has all zeros in its data buffer and waits a specified number of Data Clock Pulses before sending its data. The number of bits it waits is controlled by a constant (WAIT.BITS) in its program and must be divisible by eight.

After transmission, the microprocessor reads the count and assembles the next Data Word. This means that the data sent was taken most of one data cycle earlier.

The number and meaning of the data bits is described in Appendix II. The Data Count is the reading from the count accumulator. The Command Word is one less than the number of the command currently being executed. The Wavelength is a count of the number of steps sent to the wavelength motor since the last fiducial. The status bits are readings of fiducials and settings of relays.

The Error Bit is set by the microprocessor whenever it has difficulties. This bit will be set when:

.1 More than one command Bit is set.
.2 The Data count exceeds 13 bits.
.3 The Not Sun Present signal is active.
(even if it is overridden)

3.3.2 Motors

The microprocessor can issue step commands to the Wavelength Scan Motor, the Dust Cover Motor, the Platform Scan Motor, and the Platform Latch Motors. A count of the number of steps that have been issued since the last fiducial was seen is kept.

Each of the motors may be commanded to a Step or Hold power mode.

The microprocessor can command the shutter to close. Removing this command allows it to open.

3.4 Initiation

Any time the power is turned on, or the Reset button in the RS-232 development cable is pressed, the following actions are taken:

.1 The Microprocessor is reset.
.2 The Forth language is initiated.
.3 The Data interrupt is disabled.
.4 Memory is checked for Flight or Development
.5 The Outputs are cleared.
.7 Data Counter is cleared.
.8 Data Taking is enabled.
.9 COM.SER mode is entered.

The system is then ready to send data and respond to Spacecraft commands.

3.5 Spacecraft Commands

There are eight (8) Spacecraft commands:

1 - SCAN1	(Wavelength scan 1)
2 - SCAN2	(Wavelength scan 2)
3 - SCAN3	(Wavelength scan 3)
4 - NSP.OV	(Not Sun Present Override)
5 - FCAL	(Flight Calibration)
6 - STAN.ST	(Standard Start Position)
7 - OP.LAT	(Open Platform Latch)
8 - SHT.DW	(Normal Shut Down)

3.5.1 COMM.SER

In the COM.SER (Command Service) mode, the microprocessor card checks to see if the Sun Present circuit is tripped and then reads the Spacecraft Commands.

If a Spacecraft command is detected, the command is reread to guard against glitches, and if the readings are the same the appropriate mode is called. If there is required action for the specific command, the microprocessor remains in the COM.SER mode.

After a mode sequence is completed execution returns to the COM.SER mode.

3.5.2 SCAN1

The SCAN1 (Wavelength Scan One) Mode is the primary data taking mode. This mode is started by insuring that the instrument is in the standard start configuration with the wavelength and platform at fiducials. If the motors are not at fiducial then they are stepped there.

Program control then starts executing a chart written in software for this command. The chart consists of a list of Forth words which describe specific actions (step to a specified wavelength, scan platform, etc.). Anything the instrument can do can be entered in this chart and the primary effort between flight is to edit the chart to a new configuration.

The elasped number of data periods and the number of steps issued to each motor from its last fiducial are kept and may be used by words in the chart. The step rate of the motors may be increased during flyback to increase the fraction of time for data taking.

If the instrument has two microprocessor cards, the Scan Complete line from the Secondary Microprocessor may be read by the Primary unit and a Continue Scan signal issued only when both units have completed a scan. A timeout is included to insure that this wait does not lock up the entire system. If there are two or more Secondary Microprocessors, their Scan Complete signals are AND'ed on the Spacecraft Communication card so the Primary will know when all have completed a scan.

3.5.3 SCAN2

SCAN2 (Wavelength Scan Mode 2) is very like SCAN1 except that the chart is different.

3.5.4 SCAN3

SCAN3 (Wavelenth Scan Mode 3) again is very like SCAN1 except that the chart is different.

3.5.5 NSP.OV

NSP.OV (Not Sun Present Override) allows the safety actions taken by the Sun Present circuit to be overridden in case the Sun Sensor is defective. The Sun Present circuit is hardwired to the control of the power of the Integrated Detector and the Shutter so that the safe mode may be entered without action of the microprocessor. The microprocessor overrides this circuit so that data taking can continue. The error bit is not reset.

3.5.6 FCAL

FCAL (Flight Calibration Mode) runs a test lamp sequence while in flight. The instrument is brought to the standard start condition and execution moves to this command's chart.

The chart is like the one for SCAN1 but includes commands to control the Cal lamp, wait for the lamp to warm-up, and close the dust cover.

The platform motor is not stepped and its latch is not affected.

Upon completion the test lamp is turned off and the Dust Cover opened.

3.5.6 STAD.ST

STAD.ST (Standard Start command) sets the instrument into the configuration for the start of normal operation. This includes:

.1 Wavelength motor at fiducial.
.2 Dust Cover Open.
.3 Shutter Open.

.4 Cal Lamp OFF.
.5 Not Sun Present circuit Active.

The Scan Platform is not affected.

3.5.7 OP.LAT

OP.LAT (Open Latch command) opens the scan platform latch and brings the scan platform to its fiducial.

3.5.8 SHT.DW

SHT.DW (Shut Down command) shuts down the instrument in the normal way. This leaves :

.1 Wavelength at fiducial.
.2 Platform at fiducial.
.3 Shutter open.
.4 Test Lamp off.
.5 Latch closed.

The Not Sun Present Circuit is not affected.

3.6 Development and Interflight Reprogramming

The flight program is maintained on disk in a separate IBM PC compatible microcomputer, the Software Development System. A Forth editor is used to enter and edit the flight program into the development computer and to save it on disk. Math and logic section of the flight program may be tested and debugged directly on the development microcomputer.

A utility program is then used to down-load the flight program into the Flight instrument's development RAM. After down-loading, the development computer becomes a terminal for the Flight instrument. The program can then be run and debugged.

When problems are found, the flight instrument is stopped and the development editor run to fix the problem. This procedure allows the flight program to be tested in small blocks and problems are quickly eliminated.

When the program is complete and tested, it is burned into ROM. Two utility programs, one in the flight instrument and one in the development system, work together to first set the Autostart and restart on error flags and then upload the application program from the development RAM to the development computer. The program is in the form of ASCII characters. The development utility stores the program on disk and retransmits it to the PROM burner.

The burner ROM may then be installed into the Flight instrument and the development RAM removed. The program can then be retested as an autostarting program in ROM. The flight program may send status information out the RS-232 line to the development computer but it will ignore all incoming characters and cannot be interrupted from the keyboard.

To make interflight changes the same procedure is followed. The development RAM is reinstalled and powered through the RS-232 cable. The flight program is brought on the development computer's editor and any changes or additions are made. The program is then down-loaded and tested. When it is properly functioning it is burned into ROM and the new ROM installed in place of the old.

3.6.1 Set Up

Development or modification of the flight program requires:

.1 Special RS-232 Cable with terminating box
.2 Development and Power Supply Box
.3 PROM Burner
.4 Development Microcomputer with RS-232 port
.5 Development Utilities
    .1 Down-Load
    .2 Terminal Emulator
    .3 Up-load and Burn ROM

The development computer is connnected to the flight instrument via the Control Box and RS-232 cable. The control box allows the development computer to send RS-232 serial data either to the flight instrument or ot the PROM burner. The special RS-232 cable contains the three RS-232 conductors, a pair of wires to enable the development RAM in the flight instrument, and a pair of wires to reset the Flight instrument microprocessor. The control bos also provides +-12 volts for the RS-232 transmitter.

The RS-232 cable terminates in a small box which is in turn connected to the flight instrument by a short cable. This box contains a few components to convert the RS-232 +-12 volt signals to 0 to +5 volts. Placing these components in a small development box allows them to be removed with the development cable and thereby not flown.

The RS-232 cable should be less than 50 feet but can be long enough to allow the flight instrument to be maintained in a separate clean room.

3.6.2 Step By Step

The flight and ground programs may be edited and re-entered by the following step by step procedures.

3.6.2.1 Editing Programs

1. Make a Copy

First make a copy of the program to be edited under a new name. This may be done on the original or a new disk. The original program should not be changed directly but kept for reference. The copy may be made using the COPY command in MS-DOS :

```
A: COPY OLD-NAME.SCR B:NEW-NAME.SCR
```

Note that the extension .SCR is required by Forth and that on a new disk the Forth utility set having the extension .COM will be needed. You should also prepare a back-up disk for your program.

2. Boot Forth

Boot Forth with the new program

```
FORTH   NEW-NAME
```

The Forth initiation display will appear on the monitor. Be sure that the program shown in all lower case letters is the program you wish to edit. If the name is not correct then the correct file would not be found. Check your disk and try reentering the name by USING NEW-NAME.

3. Enter Editor

```
EDIT
```

The editor display will appear on the monitor.

4. Set automatic Date

```
S
```

(Enter your Initials)

5. Edit a Screen

To edit a particular screen:

E

(Screen Number)

Check the editor section of the Forth documents to see the various editor commands available.

6. Leave Editor

To leave the editor press Esc twice. The edited program will automatically be written to disk.

7. Printout

To print out a copy of the entire new program:

0 ?SCREENS SHOW

8. Leaving Forth

To leave Forth:

BYE

3.6.2.2 Down Loading the Flight Program

1. Correct Program

Check that the program you wish to down-load is on the active disk and the program DW.COM is also on that disk.

2. Check Hardware

Check that the power is on to the flight microprocessor and that the development cable is attached. The Development RAM must be installed and the memory switch on the development box in the RAM setting.

3. Reset Flight Microprocessor

Press the reset button on the RS-232 control box.

4. Set Baud Rate

Set RS-232 Baud rate. This must be redone after power up or after using any program which effects the COM1 command of the IBM (some word processors). The program MODE must be available on the disk in the active drive.

A: MODE COM1:48,N,7,2

5. Down Load Program

Enter

DW.COM NEW-NAME.SCR

Note that the extended .SCR must be entered. The monitor screen will clear and the message "Serial Transmission" appear.

6. Watch for Errors

The program is being transmitted and will appear on the screen without comments or blank lines. Watch for Error messages which may mean that there is a mistake in the program. The Error messages are :

	Dec	Hex	
MTG	0	0	Word is not in Dictionary. Could be bad number.
MTG	1	1	Empty Stack. A Word expected something on stack.
MTG	4	4	Isn't Unique. Not always a mistake.
MTG	8	8	Disk error. Try again.
MTG	17	11	Compilation only. use in definition.
MTG	19	13	Conditional Not paired.

Usually THEN missing.

MTG 20   14   Definition Not Finished.

Usually ; missing.

Other message numbers are not supported and should not appear.

Note the word in which the first error message occurs so that you can return to the editor and make the correction. If a word contains an error it will not be entered in the dictionary and all future references to it will cause additional errors. Usually you will be able to correct only the first one or two errors.

3.6.2.3 Burning the ROM

To burn a flight program into ROM. reference the following steps:

1. Power up the Flight Instrument.
2. Load and test the flight program from development RAM.
3. Disk drive A must have a disk with the utility UP.COM on it and room to store the flight program in hex.
4. The AMS PROM burning program BIPOLAR.EXE must be on the disk in drive A or drive B.
5. The Universal PROM Program socket box must be connected to the correct card in the Development Computer.
6. Up-load the program with the command:

UP.COM FILE-NAME.HEX

The exact contents of the development RAM will be stored in the file.

7. Run the ROM burning program:

B:BIPOLAR

8. Enter I/O Address

0200

9. Select Signetics group:

S

10 Select [J]82S191] chip:

J

11. Check that chip is blank:

C

Place chip in socket 6 and hit return.

12. Write program to ROM:

W
FILE-NAME.HEX
Offset Address 0
13. Verify program:

V

14 Write second 2K if necessary.

W
FILE-NAME.HEX
Offset Address 400
15 Verify second Chip

V

(End of Software Section)

Appendix I

MEMORY AND I/O MAPS
FLIGHT INSTRUMENT
JOB 612 SACS

J. T. Riley
06/16/86

File: XMAP612
DISK: 612 Documents

INTRODUCTION

The memory and I/O are mapped for a 8085 Forth flight instrument.

1.0 Memory Map

1.1 Memory Block Types

The memory is made up of the following memory blocks:

All numbers are in Hex.

Start	End	Type	Chip	Description	Chip Select
0000	07FF	PROM	82S191	Forth Core (2k)	0
0800	0FFF	PROM	82S191	Forth Core (2k)	1
1000	17FF	PROM	82S191	Forth Core, Assembler (2k)	2
1800	1FFF	PROM	82S191	Program (2k)	3 Flight
2000	27FF	PROM	82S191	Program (2k) (Note 1)	4 Flight
1800	27FF	RAM	HM6264	Temporary RAM (4k) (Note 1) (remove before flight)	3 4 Devel
2800	28FF	RAM	8155H	Flight RAM in port (1/4k)	5
3000	30FF	RAM	8155H	Flight RAM in port (1/4k)	6
3800	38FF	RAM	8155H	Flight RAM in port (1/4k)	7

Note 1. Either the PROM or RAM may be active during development but not both. The choice is controlled through a dedicated pair in the RS-232 cable.

2.2 Assigned Memory Locations

Individual memory locations having specific designations defined by hardware or Forth requirements are:

Add	Mnem	Description
0000	RST0	Cold start or Reset vector (sent to CLD)
0008	RST1	NOT USED (sent to WRM)
0010	RST2	NOT USED (sent to WRM)
0018	RST3	NOT USED (sent to WRM)
0020	RST4	NOT USED (sent to WRM)
0024	TRAP	TRAP interrupt vector, NOT USED (Sent to CLD)
0028	RST5	NOT USED (sent to WRM)
002C	RST5.5	Incoming RS-232 Character (sent to RST5.5)
0030	RST6	NOT USED (sent to WRM)
0034	RST6.5	NOT USED (sent to WRM)
0038	RST7	NOT USED (sent to WRM)
003C	RST7.5	Data Interrupt (sent to DATA vector)
0040		Start of Boot-up Literals (34 bytes)
0062		Start of Core and Assembly words (about 6k bytes)
	1FFF	End of Core space
1800		Vector to flight autostart Word (2 bytes)
1803		Vector to Incoming RS-232 character Word (3 bytes)
1806	18XX	FORTH Vocabulary
18xy		First Application Word (HERE .)
	27FF	End of Application Word area (about 4k bytes)
2800		Incoming Character buffer, KEY (2 bytes)
2802		Word Buffer Data Buffer (84 bytes)
2842		Text Buffer, PAD, OUT (64 bytes)
2882		User Area Pointer UP (2 bytes)
2884		User Variables (48 min. 102 bytes allotted)
3000		Temp Input Buffer TIB Data Buffer (64 Bytes)
30FF		Parameter Stack, SO (170 min. 192 bytes allotted, grows down)
37FC		Return Stack, RP. IN (48 min. 254 bytes allotted, grows down)
37FE		Return Stack Pointer RPP (2 bytes)

2.0 I/O Map

All chips are 8155H (port with RAM and timer/counter).

All numbers are in Hex.

```
I/O ADD Port   I/O     Description

PORT 2    CHIP  U10

29       A5 (8) O   Data Out (8)

2A       B2 (8) I   Fiducials , or Analog In
                          1 - Wavelength Cam, course, Fid-0
                          2 - Wavelength Motor, fine, Fid-0
                          4 - Dust Cover Open, Fid-1
                          8 - Dust Cover Closed, Fid-1
                         10 - Scan Platform Cam, course, Fid-0
                         20 - Scan Platform Motor, fine, Fid-0
                         40 - Latch Open, Fid-1
                         80 - Latch Closed, Fid-1

                 or
                    Analog In (4), Misc. (4)
                          1 - Sun Intensity Level 1, 1-High
                          2 - High Voltage Level 1, 1-High
                          4 - Temperature Level 1, 1-High
                          8 - One of above, Level 2, 1-High
                          10 - Optical Test Sensor, 1-On
                          20 - Output Data Counter, 1-MOD 8
                          40 - NOT Sun Present, 1-Normal
                          80 - Scan Complete Input, 1-Complete

2B       C5 (6)  O  Fid / Analog Select  0-ANALOG/1-FID
                    Spare (5)

Timer/counter 5     Data Out counter (/8 Pulses)
```

PORT CHIP 6

```
31        A6  (8)  I    Data In Low Byte (8)

32        B6  (8)  I    Data In High Byte (8)

33        C6  (6)  O    1 - Data In Count Enable Ø-Not/1-Enable
                        2 - Data In Counter Reset Ø-Reset/1-Count
                        Motor Control Lines (4), slow pulse
                              4 - Step Scan Motor
                              8 - Step Dust Cover Motor
                             1Ø - Step Platform Motor
                             2Ø - Step Latch Motor

Timer/counter 6        Sun Present Time out ( 1 min Pulses)
```

PORT CHIP 7

```
39        A7  (8)  I    Command In (8)

3A        B7  (8)  O    Relay Controls (8), Ø1Ø Pulse, 1 msec.
                              1 - Spectrometer motor, Hold
                              2 - Spectrometer motor, Step enable
                              4 - Dust Cover Motor, Hold
                              8 - Dust Cover Motor, Step enable
                             1Ø - Platform Motor, Hold
                             2Ø - Platform Motor, Step enable
                             4Ø - Latch Motor, Hold
                             8Ø - Latch Motor, Step enable

3B        C7  (6)  O    1 - Cal Lamp Relay, Ø - Off/ 1 - On
                        2 - Dark Shutter Optical Isolator,
Ø-OPEN/1-CLOSED
                        4 - Sun Inhibit Override
                              Ø-Inhibit enable/1-DET POWER ON
                        8 - Data Out Buffer, Pulse - Load Data
                        1Ø - Scan Complete Output
                              Ø-normal/1-Scan Complete
                        2Ø - Command Reset, Ø-Hold/1-Reset

  Time/counter 7        Slow Clock ( 1ØØØ Hz Sq. Way.)
```

3.0 USER Variables

Off Add	Description	
00	Not Used	
02	Not Used	
04	Not Used	
	Start of Fixed Order Words	
06	S0 Address	
08	R0 Address	
0A	TIB Address	
0C	WIDTH (max) of word name	
0E	WARNING number of error	
10	FENCE	
12	DP Address	
14	VOC-LINK	
	End of Fixed Order Words	
16	BLK (Disk word not used)	
18	IN (Disk word not used)	
1A	OUT (used by emit)	
1C	SCR (disk word not used)	
1H	OFFSET (disk word not used)	
20	CONTEXT vocabulary	
22	CURRENT vocabulary	
24	STATE compiling/executing	
26	BASE of numbers printed	
28	DPL output format	
2A	FLD output format	
2C	CSP error check	
2E	R# (editing not used)	
30	HLD output format	
	Start of Application USER Variables	
32	ELAP.T	Elasped time in data periods.
34	PRES.COM	Present command being executed.
36	WAVE	Wavelength step.
38	ANG	Scan Platform Angle step.
3A	ERR	Error Flag.
3C	available	
.		
.		
.		
64	Last available USER Variable	

Appendix II

SACS DATA WORD BIT IDENTIFICATION

J. T. Riley
Ø5/28/86
File: APPEN612

Introduction

The data from the instrument is in the form of a large serial data word. The size of the data word is defined for likely configurations and the meaning of each bit given.

1.Ø Word Length

The length of the data contributed by each microprocessor module must be divisible by eight. The single spectrometer system would require the following bits:

Spectrometer	
Count	13
Wavelength	1Ø
Command	3
Status	6
Error	1
Total	33
Scan Platform	
Scan Angle	13
Status	2
Total	15

The proposed configurations would then have the following word lengths.

One Spec without Scan

Spec	33
Fill	7
Total	40

One Spec with Scan

Spec	33
Scan	15
Total	48

Two Spec. with Scan

Spec 1	33
Scan	15
Spec 2	40
Total	88

Four Spec. no Scan

Spec 1	40
Spec 2	40
Spec 3	40
Spec 4	40
	160

2.0 Bit Identification

The data bits have the following identities. There is no difference between the Primary and Secondary microcomputer data.

2.1 Spectrometer.

First 8 bits

1	Count, low word, bit 1, LSB
2	Count, low word, bit 2
3	Count, low word, bit 3
4	Count, low word, bit 4
5	Count, low word, bit 5
6	Count, low word, bit 6
7	Count, low word, bit 7
8	Count, low word, bit 8

Second 8 Bits

9	Count, high word, bit 9
10	Count, high word, bit 10
11	Count, high word, bit 11
12	Count, high word, bit 12
13	Count, high word, bit 13, MSB
14	Command word, bit 1
15	Command word, bit 2
16	Command word, bit 3

Third 8 bits

17	Wavelength, low word, bit 1, LSB
18	Wavelength, low word, bit 2
19	Wavelength, low word, bit 3
20	Wavelength, low word, bit 4
21	Wavelength, low word, bit 5
22	Wavelength, low word, bit 6
23	Wavelength, low word, bit 7
24	Wavelength, low word, bit 8

Fourth 8 Bits

25	Wavelength, high word, bit 9
26	Wavelength, high word, bit 10, MSB
27	Wavelength Fiducial. 1 = At fid
28	Cover Status, 1 = Closed
29	Shutter Status 1 = Closed
30	Test Lamp Stat, 1 = On
31	High Voltage Status, 1 = On
32	Test Lamp Status, 1 = Sun Present

Fifth 8 Bits

33	Error Indicator, 1 = On
34	Fill
35	Fill
36	Fill
37	Fill
38	Fill
39	Fill
40	Fill

2.2 Scan Platform

If the scan platform is present, then the data for the Primary microporcessors has the scan information added to the data.

Fifth 8 Bits

33	Angle, low word, bit 1, LSB
34	Angle, low word, bit 2
35	Angle, low word, bit 3
36	Angle, low word, bit 4
37	Angle, low word, bit 5
38	Angle, low word, bit 6
39	Angle, low word, bit 7
40	Angle, low word, bit 8

Sixth 8 Bits

41	Angle, high word, bit 9
42	Angle, high word, bit 10
43	Angle, high word, bit 11
44	Angle, high word, bit 12
45	Angle, high word, bit 13, MSB
46	Scan Angle Fiducial, 1 = At fiducial
47	Latch Status, 1 = Latched
48	Error Bit, 1 = Error.

```
Screen # 0
( FLIGHT PROGRAM  JOB 612                          JTR 11:54 06/02/86 )
( Last change:   Screen  027                       JTR 10:33 06/16/86 )


( APPENDIX III -               FLIGHT PROGRAM                         )
(                                                                     )
(                              SACS   Job 612                         )


( Version 1.0                                                         )
( J. T. Riley                                                         )
( RSI                                                                 )



( File:  FP.SCR          Disk:  SACS    JOB 612   PROGRAM            )


Screen # 1
( USER VARIABLE, VARIABLE,  CONSTANT                   15:13 06/02/86 )
FORTH DEFINITIONS   HEX

32  USER ELAP.T            ( ELAPSED TIME, data periods )
34  USER PRES.COM          ( Present mode command )
36  USER WAVE              ( Wavelength motor step count )
38  USER ANG               ( Platform motor step count )
    USER ERR               ( Error flag  )

2802 CONSTANT DAT.BUF      ( Address of data buffer  )
125  CONSTANT MOT.F        ( Stepper Motor Frequency, Hz )
10   CONSTANT W.RELAY      ( Length relay pulse, msec )



DECIMAL


Screen # 2
( INIT.PORTS                                       JTR 11:50 06/02/86 )
ASSEMBLER  HEX

CODE  INIT.PORTS    (  --  , Initiate Ports )
      08 A MVI  2C OUT     ( Timer 5   /8 )
      CO A MVI  2D OUT     ( Timer 5 cont Pulses )
      CD A MVI  28 OUT     ( A5-O, B5-I, C5-O )
      E8 A MVI  34 OUT     ( Timer 6  /1000 )
      C3 A MVI  35 OUT     ( Timer 6 cont Pulses )
      CC A MVI  30 OUT     ( A6-I, B5-I, C5-O )
      03 A MVI  3C OUT     ( Timer 7  /3072 )
      CC A MVI  3D OUT     ( Timer 7 cont Pulses )
      CE A MVI  38 OUT     ( A7-I, B7-O, C7-O )
      NEXT JMP C:

DECIMAL
```

```
Screen # 3
(        I.VAR                                            JTR 15:16 05/30/86 )


: I.VAR    ( --   , Initialize Variables )
        0 ELAP.T          ( Zero Elapsed Time )
        0 PRES.COM        ( Present Command )
        0 ERR ! ;         ( Error flag )


DECIMAL




Screen # 4
( B5@, COM@, C7@                                          JTR 12:15 05/30/86 )
HEX
CODE B5@ (   -- B, Read   Port  B5 )
        2A     IN       ( Read Port B5 )
        L  A   MOV      ( LSB )   H  0  MVI    ( MSB )
        HPUSH JMP  C; ( Place on stack )

CODE COM@ ( -- N , Read Command input, Port A7 )
        39     IN       ( Read Port A7)
        L  A   MOV      ( LSB )   H  0  MVI    ( MSB )
        HPUSH JMP  C; ( Place on stack )

CODE C7@ ( -- N , Read  Port C7 )
        3B     IN       ( Read Port C7 )
        L  A   MOV      ( LSB )   H  0  MVI    ( MSB )
        HPUSH JMP  C; ( Place on stack )


Screen # 5
(       D!, F/A!                                          JTR 10/32 06/16/86


CODE D!  ( B  -- , Data byte out, Port A5 )
        H    POP  A  L  MOV     ( Get From Stack )
        29 OUT   NEXT JMP  C; ( Out Port A5 )

CODE F/A! ( N -- , Select Fid or Analog, Port C5 )
        H    POP  A  L  MOV     ( Get From Stack )
        2B OUT   NEXT JMP  C; ( Out Port C5 )
```

```
Screen # 6
( C6!, MR!, C7!                                    JTR 11:00 05/30/86 )

CODE C6! ( N -- , Output Port C6 )
        H    POP  A  L  MOV      ( Get From Stack )
        33 OUT   NEXT JMP   C; ( Out Port C5 )

CODE MR! ( N -- , Output Motor Relay, Port B7 )
        H    POP  A  L  MOV      ( Get From Stack )
        3A OUT   NEXT JMP   C; ( Out Port B7 )

CODE C7! ( B -- , Output Port C7 )
        H    POP  A  L  MOV      ( Get From Stack )
        3B OUT   NEXT JMP   C; ( Out Port C7 )


Screen # 7
( FID@, ANA@                                       JTR 12:22 05/30/86 )

: FID@      ( -- N , Read Fiducials )
        1 C5!     B5@ ;

: ANA@      (   -- N , Read Analog Levels )
        0 C5!     B5! ;

: T.L? ( -- F , Test Lamp Status, 0 Off, T On )
        ANA@ 10 AND :

: O.D.C.? ( -- F , Output Data Counter, 0-not/8. T-/8 )
        ANA@ 20 AND :

: SUN? ( -- F , Sun Present , 0-Save, T-Alarm )
        ANA@ 40 AND ;


Screen # 8
( ST.MOT, MOT.WAIT, W.MOT, DC.MOT                  JTR 11:52 05/30/86 )
HEX
: ST.MOT   ( N --   , Step motor # once )
        C6!  1 MSEC 0 C6! ;

: MOT.WAIT  ( --   , Wait one motor cycle )
        1 MOT.F /  1000 * 1 -   MSEC ;

: W.MOT    ( N --   , Step wavelength motor )
         0 DO
            4 ST.MOT  MOT.WAIT
        LOOP ;

: CC.MOT     ( N --   , Step Dust Cover motor )
         0 DO  8 ST.MOT  MOT.WAIT
        LOOP ;
```

```
Screen # 9
( SP.MOT, L.MOT                                    JTR 11:52 06/02/86 )

: SP.MOT   ( N --    , Scan Platform motor )
         0 DO  10 ST.MOT  MOT.WAIT
        LOOP ;

: L.MOT    ( N --    , Latch motor )
         0 DO  20 ST.MOT   MOT.WAIT
        LOOP ;

: W.HS     ( F --    , Wavelength motor, 0 Hold, 1 Step)
        1 AND  1 +  B7! ;
        W.RELAY MSEC 0 B7! :




Screen # 10
( DC.HS, SC.HS, LC.HS                              JTR 12:22 05/30/86 )

: DC.HS      ( F --   , Dust Cover motor, 0 Hold. 1 Step)
        4 AND  4 + B7!
        W.RELAY MSEC 0 B7! ;

: SC.HS      ( F --   , Scan Platform motor, 0 Hold. 1 Step)
        10 AND  10 + B7!
        W.RELAY MSEC 0 B7! ;

: LC.HS      ( F --   , Latch motor, 0 Hold, 1 Step)
        40 AND  40 + B7!
        W.RELAY MSEC 0 B7! ;




Screen # 11
( S.C.I?                                           JTR 14:05 05/30/86 )

: S.C.I? ( -- F , Scan Complete :Input, 0-No, T-Complete )
        ANA@ 80 AND ;

: T.L    ( F -- , Test Lamp Command, 0-Off, T-ON )
        1 AND C7@ FE AND +   C7! :

: SHUT   ( F -- , Shutter Command, 0-Open, T-Closed )
        2 AND C7@ FD AND +   C7! :

: D.O.B.L  (   -- , Data Out Buffer Load, Pulse )
        4  C7@   +  C7!    ( Set bit )
        C7@ FB AND C7! ;  ( Clear Bit )
```

```
Screen # 12
( S.C.OUT, COM.RS                               JTR 15:29 05/30/86 )

: S.C.OUT  ( F -- , Scan Complete Out, 0-Not, T-Complete )
       10 AND C7@ EF AND +   C7! ;

: COM.RS   (  -- , Command Reset )
       20  C7@  +  C7!    ( Set bit )
       C7@ DF AND C7! ;  ( Clear Bit )

: WFID?    (  -- F , At Wavelength Fid, T-at fid )
       FID@ 3 AND 3 =  ;   ( Two Fids )

: PFID?    (  -- F , At Platform Fid, T-at fid )
       FID@ 30 AND 30 =  ;   ( Two Fids )



Screen # 13
( NEW.COM?, ST.WFID, ST.SPFID                    JTR 15:31 06/02/86 )

: NEW.COM?  (  -- F,  New Command ? , 1 NEW )
       COM@
       PRES.COM @ =  NOT ;  ( Does not change command )

: ST.WFID   ( --   , Step to Wavelength Fiducial )
       BEGIN WFID? NOT WHILE  ( At fid ? )
          1 W.MOT REPEAT ;    ( Step to fid )

: ST.SPFID   ( --   , Step to Scan Platform Fiducial )
       BEGIN SPFID? NOT WHILE  ( At fid ? )
          1 SP.MOT REPEAT ;    ( Step to fid )



Screen # 14
(                                               JTR 15:36 06/02/86 )

: OP.DC ( -- , Open Dust Cover with time out )
    TRUE DC.HS
    200 0 DO 1 DC.MOT ;
```

```
Screen # 15
( Additional command words                        JTR 11:16 04/09/86 )
```

```
Screen # 16
( STAN.ST                                         JTR 10:32 06/16/86 )

: STAN.ST  (  --    ,  Standard Start Condition )
                                   ( Step to Wave Fid )
                                   ( Step to Angle Fid )

        ;
```

```
Screen # 17
(                                                 JTR 11:11 05/30/86 )
```

```
Screen # 18
(                                                    JTR 11:11 05/30/86 )
```

```
Screen # 19
(                                                    JTR 11:11 05/30/86 )
```

```
Screen # 20
(                                                    JTR 11:11 05/30/86 )
```

```
Screen # 21
(                                                  JTR 11:12 05/30/86 )
```

```
Screen # 22
(                                                  JTR 11:12 05/30/86 )
```

```
Screen # 23
(                                                  JTR 11:12 05/30/86 )
```

```
Screen # 24
 SER.COM.                                             JTR 11:13 05/30/86 )

 SER.COM   (  --   , Service incoming commands )
   BEGIN  1 WHILE           ( Infinite loop )
       COM.CK               ( Get new command )
           DUP 1 = IF , EARTH EXECUTE
           DUP 2 = IF , SUN   EXECUTE
               4 = IF , CAL   EXECUTE
       STANDBY
       SUN?
       SRU
   REPEAT ;
```

```
Screen # 25
( CT@                                                 JTR 14/30 05/30/86 )

CODE CT@ ( -- N , Read Detector count, Port A6, B6  )
        A  0  MVI
        33    OUT      ( Stop Count )
        31    IN       ( Read Port A6, LSB )
        L  A  MOV      ( LSB )
        32    IN       ( Read Port B6 )
        H  A  MOV      ( MSB )
        A  2  MVI
        33    OUT      ( Reset Counter )
        A  1  MVI
        33    OUT      ( Enable Counter )
        HPUSH JMP  C;  ( Place on stack )
```

```
Screen # 26
( DATA@                                               JTR 11:54 06/02/86 )

: DATA@  ( --   , Read new Data, Store at DAT.BUF )
        CT@ DUP E000 AND  IF 1 ERR ! 1FFF AND  ( Count )
        PRES.COM 2000 * + DAT.BUF !       ( Command )
        WAVE 03FF AND                     ( Wavelength )
        WFID? 4000 AND SWAP                ( Wave Fid )
        8 AND 200 * +                      ( D C status )
        SHUT  1000 +                       ( Shutter )
        ANA@ DUP DUP 40 AND 80 * SWAP      ( Sun present )
        2 AND 2000 * + SWAP                ( High Voltage )
        10 AND 800 * + +   2 DAT.BUF + !   ( Test Lamp )
        ANG 1FFF AND                        ( Platform Angle )
        PFID? 4000 AND SWAP                 ( Platform Fid )
        8 AND 800 * + +                     ( Latch Status )
        ERR 8000  AND  + 4 DAT.BUF + ! ;    ( Error )
```

```
Screen # 27
( DATA, Data handling in Forth                        JTR 10:33 06/16/86 )

: DATA    (    --    , Data service on interrupt 7.5   )
        49 0 DO
           BEGIN  NSP@ 2 AND NOT WHILE ( Wait for counter )
           DAT.BUF 49 I - + C!   D.OUT   ( Output data byte )
        LOOP
        DATA@                     ( Read new Data   )
        DAT.BUFF @ D.OUT          ( Preload output )
        1 ELAS.T +!               ( Increment elasped time   )
        IRET ;                    ( End interrupt )



Screen # 28
( INT7.5                                              JTR 11:57 03/31/89 )

CODE INT7.5   ( -- ,  Set up Forth interrupt 7.5 )
        PSW PUSH B PUSH D PUSH H PUSH
        H   , DATA 2 - 256 /     MVI     ( Code Field Address )
        L   , DATA 2 - 256 MOD   MVI
        E     M      MOV                 ( code address )
        H            INX
        D     M      MOV
        XCHG                             ( Code Address in HL )
        PCHL                             ( Mimic   NEXT )
        C:



Screen # 29
( Initiate Interrupt 7.5. INIT                        JTR 11:17 04/09/86 )
HEX
        C3      1802 C!                  ( JMP command )
        '       INT7.5 2 -   1803   !   ( Upper interrupt Vector )

: INIT    (   --    , Initiate System )
        I.COM  I.VAR  INIT.PORTS      ( Initiate system )
        DI   0 MFP!   OFF TAPE SRU SUN?
        1000 0 DO LOOP                  ( Wait )
        ON MFP!  EN.I                   ( Enable Data )
        SER.COM ;           ( Service incoming Commands )

        '       INIT    1800 !    ( Cold Start vector )

( End of Program )
HERE 27FF SWAP - DECIMAL CR ." Memory remaining - " . CR


Laboratory Microsystems PC/FORTH 3.00       11:02   06/16/86   fp.scr
```

APPENDIX F

ELECTRONICS MANUAL

FLIGHT INSTRUMENT

HUP2

JOB 612

Contract No. F19628-79-C-0044

Research Support Instruments, Inc.
10610 Beaver Dam Road
Cockeysville, Maryland 21030-2288
(301) 785-6250

J. T. Riley
01/05/89
DISK: HUP2 612

Files: MAN612
SOFT612
DEV612
Illustrations
MAP612
WORDS85
DUAL
MOD612
FP1.SCR
FPM.SCR
FPS.SCR

TABLE OF CONTENTS

SOFTWARE

3.Ø Software Description
3.1 General Philosophy
3.1.1 Forth Core
3.1.2 Structure
3.1.3 Development
3.1.4 Ground Testing
3.1.5 Flight Program
3.2 Input
3.3 Output
3.4 Initiation
3.5 Spacecraft Commands
3.6 Master Program Review
3.7 Secondary Program Review

Development and Reprogramming

4.Ø The Development System
4.1 Hardware
4.2 Set Up
4.3 Editing Programs
4.5 Down Loading the Flight Program
4.6 Burning the PROM

FIGURES

Appendix

ASSOCIATED PRINTS

Microprocessor Card Schematic	Roll 800-224-40-4
Dual Stepper Motor Driver, Schematic Diagram	D480-244-0-2
Schematic Diagram, Dual Motor dirver	D490-224-0-2
IBM Spacecraft Emulator Card Schematic	D910-224-0-5
RS-232 Box with Optical Isolator, Schematic	C860-224-0-2

CONTRACT CONTACT

Charles Forsberg -LIU (617) 377-2625
AFGL
Air Force Systems Command
USAF
Hanscom Air Force Base
Mass. 01731

FLIGHT INSTRUMENT ELECTRONICS

GENERAL DESCRIPTION

INTRODUCTION

This document is the primary written reference for the electronics of the HUP2 612 project. It serves as the design specification and the operating manual for both the flight instrument and the Software Development System. The hardware and software for the system are described in detail.

1.0 Purpose of Document

This document serves as the specification for the electronics portion of this project. The function of the electronic hardware and software are described in detail.

Also included are instructions for the operation of the Software Development System. These provide step-by-step instructions for the writing of software, its debugging, and the burning of PROMs.

The optics and machanical design are not covered.

1.1 Description of Flight Instrument Electronics

The flight electronics is composed of a number of printed circuit cards mounted in individual aluminum boxes and connected by cabling. This system controls the flight instrument in response to commands from the spacecraft and transmits data to the spacecraft.

1.2 Description of Software Development System

The Software Development System consists of a small computer, power supply, and cables which support development of the flight software and debugging of both the software and hardware.

HUP2 HARDWARE

INTRODUCTION

The hardware for the HUP2, a two spectrometer instrument with scanning platform, is described in detail. This same electronic system may be used with two or four spectrometers and either with or without a scaning platform. The hardware for the Software development is also described.

2.0 Hardware Description

The electronics for the HUP2 multiple spectrometer instrument is divided into the following modules:

1. EMI Filter (1 per system)
2. +5 Power Supply (1 per system)
3. Spacecraft Communications Module (1 per system)
4. Microprocessor Module (1 per spectrometer)
5. Motor Control Module (1 per two motors)
6. Motor Control Module, DC (1 per platform)
7. Integrated Detector (1 per spectrometer)

Also on the instrument are the following electronic devices:

1. Wavelength Motor (1 per spectrometer)
2. Wavelength Cam Fiducial (1 per spectrometer)
3. Dust Cover Motor (1 per spectrometer)
4. Dust Cover Fiducials (2 per spectrometer)
5. Shutter (1 per spectrometer)
6. Test Lamp (1 per spectrometer)
7. Platform Motor (1 if platform used)
8. Platform Cam Fiducials (2 if platform used)
9. Platform Latch DC Torque Motor (1 if platform used)
10. Platform Latch Fiducials (2 if platform used)
11. Sun Sensor (1 per spectrometer)
12. Flight Cables (1 set)

The system (see Figure 21) consists of the basic support equipment (EMI filter, Power Supply, Spacecraft Communications Module, and cables); a Master Microprocessor (Microprocessor Module, Motor Control Module, Integrated Detector, motor, etc.) and one or three

Secondary Modules (identical hardware with Master). If the system has a scan platform then an additional Motor Control Module will be cabled to the Master Microprocessor.

Each module will consist of a single printed circuit board mounted in a machined aluminum housing with subminiature D connectors. The components are chosen for high reliability. All components which dissipate significant heat will be heatsunk to the module cases.

2.1 Electronic Subassemblies

The following electronic subassemblies are used in the system.

2.1.1 EMI Module

The EMI module prevents electromagnetic from getting from the instrument back to the spacecraft over the power lines. It also smooths and filters the incoming +28 volt power for the instrument.

2.1.2 +5 Power Supply

The +5 power supply converts +28 volt power into regulated +5 voltage for use by the Microprocessor Modules, the Spacecraft Interface Module, The Motor Control Modules, and the Integrated Detectors. This is a standard high reliable flight power supply and its size is dependent on the number of spectrometers in the system.

2.1.3 Spacecraft Interface Module

The Spacecraft Communication Module acts as the interface between all the Microprocessor Modules and the spacecraft. It performs the following functions:

.1 Buffers 8 incoming Command Lines.
.2 Buffers incoming Data Enable and Data Clock lines with a separate output for each instrument.
.3 OR's serial data lines from all microprocessors.
.4 Buffers outgoing data.
.5 Provides +10 volt power for analog Circuits. To a Maximum of 75 milliamps per microprocessor.
.6 AND's Scan Complete signal from multiple secondary Microprocessors.

The HUP2 instrument, or instruments, may be used on a variety of spacecraft by redesigning only this module and reprogramming the microprocessor modules. If only one instrument is to be used and the spacecraft communication link is TTL compatible, then this module may not require active components. For multiple instruments the data output lines will be OR'ed together here. Some spacecraft may require the data lines to be inverted or shifted to 0 - 10 Volt levels, this type of adaptation is easily implemented with a few chips in this module.

2.1.4 Microprocessor Module

The Microprocessor Module is the brain of the instrument and contains the following sections.

2.1.4.1 Forth Core

This is a 8085 microprocessor with a Forth operating system in ROM. It allows the flight software to be developed quickly in a higher level language and supports software debugging. A detailed allocation of the memory is included in Appendix IV, Memory and I/O Map.

All the ROM, both that is used for the Forth Core and used for the Flight Program, is of the UV erasable type. A Software development RAM chip is placed in the Flight Program PROM socket during development. The choice of RAM or PROM for this socket is determined by a jumper on the microprocessor board. Provision is made on the board for either a temporary wire wrap jumper or a soldered wire. The development RAM is not flown to save power and space, and to improve reliability.

The core has a RS-232 port which is used only for software development and debugging. The board output and input are at TTL levels for simplicity and a separate box in the development cable contains a converter to the +-12 volt standard RS-232 levels.

2.1.4.2 Data In and Out

The data transmission system is initiated by the Data Enable signal tripping the 7.5 interrupt. The data is then transferred from memory to a 8 bit shift register which shifts the data out on the Data Clock pulses. The Data Clock pulses are also counted in hardware with a divide by 8 so that the timing of the transfer on the next data word is easily accomplished at high speed.

2.1.4.3 Fiducials

All the fiducials consist of LED/Phototransistor pairs mounted so that a hole in a mechanical device is aligned with them at the reference mark. Fiducials read the edges of the hole or flag and not the center of the hole. When the fiducial is aligned at its reference mark it reads a logical Ø.

The fiducials on the motors read a hole in a disk attached to the drive cam. This produces one reference mark per complete cycle and this mark is located in the middle of the flyback segment.

The dust cover and latch have fiducials at the fully open and the fully closed position.

2.1.4.4 Analog Signals

Up to four analog signals may be input, buffered, and output by the microcomputer card. The outputs are not inverted and have a 2K series resistor. Each of these levels may be compared to a reference level set by a resistor and these comparisons can be read by the microprocessor. If all four comparisons are not needed, one signal may be compared against two levels.

The selection of comparitors and power source for the analog buffer is set by jumpers on the microprocessor board. The reference voltage level for the comparators is set by resistors on the microprocessor board. The analog chips may be powered by an external power source on the Spacecraft Interface card or from the internal +5 volts.

This allows the microprocessor to read hardware defined levels or windows of the housekeeping data. This may be used to detect the correct level for the detector HV.

2.1.4.5 Scan Complete

To coordinate the wavelength scans of more than one microprocessor, a SCAN COMPLETE INPUT and a SCAN COMPLETE OUTPUT signal are provided. The secondary units set the SCAN COMPLETE OUTPUT high when they complete a wavelength scan. They then wait for the SCAN COMPLETE INPUT to go high or a time-out to occur before starting the next scan. The SCAN COMPLETE OUTPUT is reset on the start of the new scan.

The Primary microprocessor, upon completing its wavelength scan, waits for the SCAN COMPLETE INPUT to go high or for a time-out to occur before starting its next scan and pulsing the SCAN COMPLETE OUTPUT high for 1 millisecond.

For an instrument with more than one secondary microprocessor, the SCAN COMPLETE OUTPUT signals from the secondary units are AND'ed together on the Spacecraft Interface Card and the result sent to the Primary unit. This feature is not implemented if only one instrument is used.

2.1.4.6 Motor Control

The microprocessor may control up to 4 stepper motors but cannot provide enough current to drive the motors. Relay control lines are also provided to switch the motors from a step to a hold state.

2.1.4.7 NOT SUN PRESENT

The NOT SUN PRESENT signal triggers a hardware function to protect the detector from excessive sunlight. The microprocessor can sense the state of this circuit and override the safety if commanded to do so.

When the NOT SUN PRESENT signal is high, a safe and normal condition exists. The Detector Power Relay is energized so that the detector is operational. The Shutter Relay is not energized, therefore, the Shutter is open.

When the NOT SUN PRESENT signal goes low, the Detector Power Relay is de-energized so the detector high voltage is off, and the Shutter Relay is energized and the Shutter closes. A one minute timer is also started which holds this situation for at least one (1) minute. After one minute, if the NOT SUN PRESENT signal has returned to high, normal operation is resumed.

If the microprocessor receives a command 4, HV.OVR, it overrides the effects of the NOT SUN PRESENT signal and the instrument is placed in the normal operating mode. This command is used if the Solar Sensor is malfunctioning. The NOT SUN PRESENT signal is reactivated after a Command 6, Standard Start Position, is received.

If the microprocessor card is powered, but the microprocessor is not running, the NOT SUN PRESENT signal going low will lock the detector power off and it will not reset after one (1) minute.

The microprocessor may also separately energize the Shutter.

2.1.4.8 Spacecraft Commands

The microprocessor can read 8 command lines from the spacecraft. These are the same 8 lines for all microprocessors but different instruments may respond differently.

2.1.5 Motor Control Module

The Motor Control Module provides drive power for two four-phase stepper motors. The motor drivers have input buffers, pulse counter, optical isolators, and headsunk power transistor. The +28 Volt supply and return are completely isolated from the +5 Volt supply and return. Latching relays are included to switch the motors between a holding current and a stepping current.

One Motor Control Module can power the Wavelenth Scan Motor and the Dust Cover Motor which are always steppers. If the Platform Scan Motor and the Platform Latch Motor are both steppers, a second Motor Control Module of the same type can be used for them. If the Platform Latch Motor is a DC motor instead of a stepper, then a second type of Motor Control Module must be used which controls one stepper and one DC motor.

Also included in the stepper motor version of the Motor Control Box are two relays to control power to the Detector and the Cal Lamp. These relays are not latching. A Shutter driver circuit with optical isolator is also present. These features are not present in the DC motor version.

2.1.5.1 Motors

All the Wavelength, Dust Cover, and Platform are four-phase +28 VDC stepper motors. The Platform Latch Motors is a DC torque motor. The Shutter is a +28 VDC one step motor.

The Wavelength motor always goes in one direction. The Dust Cover, Platform and Latch Motors are reversible.

2.1.6 Integrated Detector

The Integrated Detector puts out a pulse for each event detected by the Photomultiplier Tube. The pulses are counted on the microprocessor card.

This device also outputs an analog high voltage monitor signal which may be sensed by the microprocessor system to determine proper operation of the Integrated Detector.

2.1.7 Fiducials

All fiducials are LED/Phototransistor pairs that put out a low signal when the light path is not interrupted which is the state at the reference mark. The high to low transition, therefore, represents the reference position.

2.1.8 Shutter

The Shutter is a 1/3 turn DC rotary solinode. It is spring loaded open and closes when a signal is applied. When energized, the Shutter blocks the entrance slit allowing the reading of dark counts. The Shutter also blocks light from the Test Lamp.

This device draws substancial power continuously when activated.

2.1.9 Sun Sensor

The Sun Sensor uses a phototransistor to sense the presence of the sun in or near the field of view of the instrument. Its output is a digital signal, NOT SUN PRESENT, which is Ø when the sun is present and 1 when the field of view is safe. This signal is used to protect the detector from excessive sunlight.

The sensitivity of the unit is set by an internal resistor which must be chosen before encapsulation. The field of view of the sensor is set by the mechanical housing which must be set before assembly.

2.2 Power Requirements

The electronic modules require the following amounts of power:

Module	+28	+5	
1. EMI Filter	0.0	0.0	Watts
2. +5 Power Supply	12.0	0.0	
3. Spacecraft Interface Module	2.0	2.0	
4. Microprocessor Module	0.0	8.0	
5. Motor Control Module	3.0	1.0	
6. Motor Control Module, DC	3.0	1.0	
7. Integrated Detector	4.0	0.0	
System (2 instruments and Platform)			
.1 Run	25.0		Watts
.2 Hold (motors hold, detector off)	15.0		Watts

All the above figures were obtained by calculating estimates from measurements of similar modules.

2.3 Software Development System

The following hardware is provided to develop software for the instrument.

Software Development

.1 RS-232 Cable
with converter and reset boxes.
.2 RS-232 power supply
.3 Test power cable.

The following equipment is required, but not provided, under this contract:

.1 IBM PC compatible computer
with RS-232 port.
.2 Printer (optional)
.3 +28 Volt bench power supply.

This equipment is shown in Figure 20.

2.4 Calibration and Testing System

The following hardware is required to calibrate and test the instrument.

.1 IBM PC compatible computer
(The development computer can be used)
.2 IBM Spacecraft Emulator Card
.3 Test Data cable.
.4 +28 Volt bench power supply
(Same one used for software development)
.5 Power Cable
(Same one used for software development)

This equipment is shown in Figure 20. Details for the used of this equipment are given in a separate manual, Calibration and Testing System using the IBM PC.

To use the same computer for software development and for calibration and testing then the calibration program must be burned into PROM before the test.

2.5 Microprocessor Board Jumpers

The following jumpers are on the microprocessor board.

2.5.1 Development RAM / Flight EPROM

This jumper controls the Not Write line to the Development Ram. When Development RAM is being used, then this jumper should be installed.

Remove this jumper before flight.

2.5.2 Case Ground / Power Return

This jumper connects the case ground to the power return line. Most systems do not use this jumper. Install only if required by your ground system.

2.5.3 Analog System Power

This jumper allows the analog chips to be powered either from the internal +5 supply or from an external power supply. Using an external supply will allow the analog system to operate over a larger range. This voltage should be about 2 volts over the highest analog level expected. If the internal supply is used, the hightest level that could be sensed is about +3.5 volts.

Install this jumper either to the internal or external supply.

2.5.4 Analog Sense Channels

Several jumpers may be installed in the analog system to choose inputs.

If all four channels are used for different signals, then no jumpers are needed.

Any unused inputs should be jumpered to ground.

If an input signal is to be sensed at more than one level, then a jumper is needed to tie the incoming signal to the additional channels.

2.5.5 Analog Sense Levels

The resistors R51 through R57 set the trip level for the four analog sense channels. These are simple voltage dividers with a 10 Kohm resistor off the selected analog supply voltage. The values may be determined by calculation or testing.

Unused channels should have 10 Kohm resistors.

2.5.6 Port 3 Counter

The counter in port chip 3 may be used if its output is jumpered to one of the analog inputs and the resistor for this channel set at 10 Kohms. This feature is not normally used.

(End of Hardware Section)

HUP2 FLIGHT SOFTWARE

J. T. Riley
04/11/88
File: SOFT612

INTRODUCTION

The software for the HUP2 microprocessor cards is first generally discussed and then details of the flight application program are given.

3.0 Software.

3.1 General Philosophy

The flight instrument software is based on a Forth language core with the application software added in its own PROM or in a temporary development RAM.

In the design phase, the application was broken down into functional blocks each with one entrance and one exit. These blocks are written as words in the Forth dictionary. The large blocks are subsequently broken down into smaller blocks and this process continued until only words in the Forth core, or new assembly language words are needed to write the blocks.

3.1.1 Forth Core

The Forth language core is burned into 8K of UV erasable PROM and provides a basic set of functions for compiling new portions of the program, interpreting the completed program, and doing math and logic. Also included is a small assembler to allow defining of assembly language words.

The Forth core is based on the FigForth model with the disk operating system removed. During development the program is stored on the disks of a separate microcomputer, therefore, no disk operating system is needed in the flight instrument.

3.1.2 Structure

The Flight program contains only words needed in flight. Other functions, such as ground calibration, can be supported by temporary programs loaded into the development RAM.

The Flight program is divided into three (3) main Words; INIT which sets up the instrument, DATA which reads and transmits data, and SER.COM which receives and services commands. Each of these words is broken down into more detailed words.

The INIT initiates the systems variables, resets all functions to the defined starting situation, and puts the instruments through the start-up sequence. This is the Word which is executed on power up, or on pressing the reset button through the RS-232 cable. When this word is completed, execution moves to SER.COM.

The DATA word is executed upon the 7.5 interrupt line which is controlled by the Data Enable signal from the Spacecraft Interface Box. This word will be run once each time a Data Enable occurs after the INIT is complete. The Data is first transmitted to the Spacecraft Interface Card as a long serial word, and then new data is read and assembled in RAM. The first word of the new data is loaded into the data buffer. At the end of this sequence the elapsed time clock is incremented and execution returned to SER.COM.

The SER.COM word is an infinite loop and may be left only by hardware reset or powering down. The data interrupt does periodically interrupt SER.COM but its activity is not affected by this. SER.COM first looks at the Spacecraft Command Inputs to determine if there is a valid new command. If there is no new command, the program simply waits rechecking the Not Sun Present signal and the incoming comand lines periodically. If there is a valid new command, the word for that command is called. The specific command words (SCAN1, SCAN2, FCAL, ect.) produce a sequence of actions such as a pattern of steps to the wavelength motor or a command to the Test Lamp. Between other functions the microprocessor checks for new incoming commands, if a new valid incoming command is found, execution of the present command is stopped, STAD.ST is run, and the new mode sequence started.

3.1.3 Development

The application portion of the software is burned into 8K of UV erasable PROM. In development, the program is written on the external computer and down-loaded into the RAM through the RS-232 port. After all necessary tests and debugging have been completed, the final flight program is transmitted up the RS-232 cable, stored on disk, and retransmitted to the PROM burner. The Flight PROM may then be inserted in the electronics module and the development RAM removed.

A number of utility words have been added to the Forth core to facilitate the development process. These down-load and up-load features assist in burning PROMs and are described in Section 4.Ø Development System.

3.1.4 Ground Testing

The flight instrument may be exercised on the ground through the RS-232 port if desired. This allows most of the instrument functions to be tested without the use of the ground support equipment.

Data transfer may not be fully tested without an additional CTS card for the development computer and additional CTS software.

3.1.5 Flight

In flight the software will:

Initiate the System
- .1 Set up the Software.
- .2 Test for Flight or Development mode.
- .3 Initiate Data reading and transmission.

Handle Data (under interrupt)
- .1 Transmit Data to the Spacecraft Interface Card.
- .2 Read Detector Event Counter.
- .3 Read Status information.
- .4 Assemble the next Data Word.
- .5 Increment the elapsed time clock.

Command the Instruments
- .1 Receive commands from Spacecraft Interface.
- .2 Issue steps to the wavelength motor.
- .3 Issue steps to the platform scan motor.
- .4 Operate the Test Lamps.
- .5 Respond to Sun Presence.
- .6 Operate Shutter.
- .7 Operate Dust Cover.
- .8 Read all Fiducials.
- .9 Operate platform latch.

.10 Coordinate with other instruments

3.2 Input

The microprocessor has available the following inputs:

.1 Spacecraft Commands (8)
.2 Fiducial Readings (8)
.3 Data Enable and Data Clock
.4 Detector Counter (16 Bit)
.5 Analog Signal Levels (4)
.6 Not Sun Present (1)
.7 Cal Lamp On (1)
.8 Scan Complete In (1)
.9 Development RS-232 port.
(Not normally used in flight)

The eight (8) Spacecraft Commands allow the microprocessor to respond to commands from the spacecraft. The software must respond to each of these with a sequence of actions.

The Fiducials give reference marks for moving parts. The eight (8) fiducials for the primary system are:

.1 Wavelength Cam Course
.2 Spare
.3 Dust Cover Open
.4 Dust Cover Closed
.5 Scan Platform Cam Course
.6 Spare
.7 Platform Latch Open
.8 Platform Latch Closed

On the secondary system or primary system without a scan platform, the Scan Platform and Platform Latch fiducials are also spare.

The microprocessor card senses the Data Enable line and responds with data clocked by the Data Clock. The microprocessor keeps count of the data interrupts in the form of a double precision number, and generates an elapsed time to a maximum of 2 billion (2 to the 32 power) data periods. This time information maybe used to start and stop actions during the command sequences if the data system is running.

The microprocessor may stop, read, and reset a 16 bit hardware counter for the Detector Pulses.

Three (3) analog signals are buffered in this box and sent to the Spacecraft Interface Card. One of these, the High Voltage monitor is read by a digital level sensing circuit so that the state of the high voltage to the detector may be read.

The Not Sun Present Signal indicates if the instrument is pointed at the sun (Ø level). Emergency action is taken by hardware, closing the shutter and turning off the high voltage to the detector. The microprocessor sets an error bit and starts a time-out counter in software. When the time-out period is over, the microprocessor resets the protection system.

The Cal Lamp On Signal indicates that the calibration lamp is on.

The Scan Complete Input indicates that the other microprocessor card, has completed a major program action.

The Development RS-232 port is used only for development and debugging software and is not normally used in flight.

3.3 Output

The microprocessor has the following outputs:

.1 Serial Data
.2 Issue Motor Steps
.3 Set Motor Power Level, Step/Hold
.4 Operate Shutter
.5 Operate Cal Lamp

.6 Scan Complete Out
.7 Not Sun Present Override/Reset

3.3.1 Data

The spacecraft sends a Data Enable Signal and a Data Clock to the Spacecraft Interface card. This card buffers these signals and sends them to all Microprocessor Cards. If multiple microprocessors are present, this card ORs the returning data signals together, buffers the resulting signal, and sends it to the spacecraft.

The Data Enable line from the Spacecraft Interface card initiates the data transfer by firing the 7.5 interrupt of the microprocessor.

If the unit is the Primary microprocessor card then it immediately issues the latest data word which it has previously stored in its memory. The data is clocked out by the Data Clock. If the microprocessor card is a secondary card, it has all zeros in its data buffer and waits a specified number of Data Clock Pulses before sending its data. The number of bits it waits is controlled by a constant (WAIT.BITS) in its program and must be divisible by eight.

After transmission, the microprocessor reads the count and assembles the next Data Word. This means that the data sent was taken most of one data cycle earlier.

The number and meaning of the data bits is described in Appendix V. The Data Count is the reading from the count accumulator. The Command Word is one less than the number of the command currently being executed. The Wavelength is a count of the number of steps sent to the wavelength motor since the last fiducial. The status bits are readings of fiducials and settings of relays.

The Error Bit is set by the microprocessor whenever it has difficulties. This bit will be set when:

.1 More than one command Bit is set.
.2 The Data count exceeds 13 bits.
.3 The Not Sun Present signal is active.
(even if it is overridden)

3.3.2 Motors

The microprocessor can issue step commands to the Wavelength Scan Motor, the Dust Cover Motor, the Platform Scan Motor, and the Platform Latch Motor. A count of the number of steps that have been issued since the last fiducial was seen is kept.

Each of the motors may be commanded to a Step or Hold power mode.

The microprocessor can command the shutter to close. Removing this command allows it to open.

3.4 Initiation

Any time the power is turned on, or the Reset button in the RS-232 development cable is pressed, the following actions are taken:

.1 The Microprocessor is reset.
.2 The Forth language is initiated.
.3 The Data interrupt is disabled.
.4 Memory is checked for Flight or Development
If in Development then the Monitor is entered.
If in Flight the Autostart program is run.
.5 The Outputs are cleared.
.6 Data Counter is cleared.
.7 Data Taking is enabled.

.8 COM.SER mode is entered.

The system is then ready to send data and respond to Spacecraft commands.

The Spacecraft may control this Reset line with a Normally open relay.

3.5 Spacecraft Commands

There are eight (8) Spacecraft commands:

1 - SCAN1	(Wavelength scan 1)
2 - SCAN2	(Wavelength scan 2)
3 - SCAN3	(Wavelength scan 3)
4 - NSP.OV	(Not Sun Present Override)
5 - FCAL	(Flight Calibration)
6 - STAN.ST	(Standard Start Position)
7 - OP.LAT	(Open Platform Latch)
8 - SHT.DW	(Normal Shut Down)

3.5.1 COMM.SER

In the COM.SER (Command Service) mode, the microprocessor card checks to see if the Sun Present circuit is tripped and then reads the Spacecraft Commands.

If a Spacecraft command is detected, the command is reread after a short time delay to guard against glitches, and if the readings are the same the appropriate mode is called.

After a mode sequence is completed execution returns to the COM.SER mode.

3.5.2 SCAN1

The SCAN1 (Wavelength Scan One) Mode is the primary data taking mode. This mode is started by insuring that the instrument is in the standard start configuration with the wavelength and platform at fiducials. If the motors are not at fiducial then they are stepped there.

Program control then starts executing a chart written in software for this command. The chart consists of a list of Forth words which describe specific actions (step to a specified wavelength, scan platform, etc.). Anything the instrument can do can be entered in this chart and the primary effort to prepare for a specific flight is to edit the chart to a new configuration.

The elasped number of data periods and the number of steps issued to each motor from its last fiducial are kept and may be used by words in the chart. The step rate of the motors may be increased during flyback to increase the fraction of time for data taking.

If the instrument has two microprocessor cards, the Scan Complete line from the Secondary Microprocessor may be read by the Primary unit and a Continue Scan signal issued only when both units have completed a scan. A time-out is included to insure that this wait does not lock-up the entire system. If there are two or more Secondary Microprocessors, their Scan Complete signals are AND'ed on the Spacecraft Communication card so the Primary will know when all have completed a scan.

The program steps used on the first instrument for SCAN1 were:

.1 Set Platform at fiducial.
.2 Set Wavelength at fiducial.
.3 Set Wavelength to 1356 Angstroms.
.4 Scan Plan 20 degrees, 1/2 complete scan.
.5 Set Wavelength to 1383 Angstroms.
.6 Complete scan.
.7 Set Wavelength at Fiducial.
.8 Run six (6) wavelength Scans while
running one Platform scan.
.9 Check for new command
If none, loop to .3 above.

All units will initially be programmed for these steps and adjustments can be made when a flight is set.

3.5.3 SCAN2

SCAN2 (Wavelength Scan Mode 2) is very like SCAN1 except that the chart is different. The program steps used on the first instrument for SCAN2 were:

.1 Set Platform at fiducial.
.2 Set Wavelength at fiducial.
.3 Scan Wavelength
.4 Check for new command.
If none, loop to .3 above.

All units will initially be programmed for these steps and adjustments can be made when a flight is set.

3.5.4 SCAN3

SCAN3 (Wavelenth Scan Mode 3) again is very like SCAN1 except that the chart is different. The program steps used on the first instrument for SCAN3 were:

.1 Set Platform to fiducial.
.2 Set Wavelength at fiducial.
.3 Set Platform to Ø degree position.
.4 Set Wavelength at 1551 angstroms
.5 Transmit data and wait for new command.

All units will initially be programmed for these steps and adjustments can be made when a flight is set.

3.5.5 NSP.OV

NSP.OV (Not Sun Present Override) allows the safety actions taken by the Sun Present circuit to be overridden in case the Sun Sensor is defective. The Sun Present circuit is hardwired to the control of the power of the Integrated Detector and the Shutter so that the safe mode may be entered without action of the microprocessor. The microprocessor overrides this circuit so that data taking can continue. The error bit is not reset.

3.5.6 FCAL

FCAL (Flight Calibration Mode) runs a test lamp sequence while in flight. The instrument is brought to the standard start condition and execution moves to this command's chart.

The chart is like the one for SCAN1 but includes commands to control the Cal lamp, wait for the lamp to warm-up, and close the Dust Cover.

The platform motor is not stepped and its latch is not affected.

Upon completion, the test lamp is turned off and the Dust Cover opened.

The command sequence for FCAL on the first instrument were as follows:

.1 Test Lamp On
.2 Dust Cover Closed, note state.
.3 High Voltage On, note state.
.4 Set Wavelength to fiducial, note position.
.5 Scan Wavelength five (5) times.
.6 Check Optical Test Indicator
In not On, call error routine, continue.
.7 Scan Wavelength five (5) times.
.8 Test Lamp Off
.9 Dust cover to orginal state.
.10 High Voltage to orginal state.
.11 Wavelength to orginal position.
.12 Check for new command.

All units will be initially programmed in this manner and they may be adjusted when a flight is set.

3.5.6 STAD.ST

STAD.ST (Standard Start command) sets the instrument into the configuration for the start of normal operation. This includes:

.1 Wavelength motor at fiducial.
.2 Dust Cover Open.
.3 Shutter Open.
.4 Cal Lamp OFF.
.5 Not Sun Present circuit Active.
.6 HV power On.

The Scan Platform is not affected.

3.5.7 OP.LAT

OP.LAT (Open Latch command) opens the scan platform latch and brings the scan platform to its fiducial.

3.5.8 SHT.DW

SHT.DW (Shut Down command) shuts down the instrument in the normal way. This leaves :

.1 Wavelength at fiducial.
.2 Platform at fiducial.
.3 Shutter open.
.4 Test Lamp off.
.5 Latch closed.

The Not Sun Present Circuit is not affected.

3.6 Master Program Custom Points

The program for the master instrument has the following special features because it is the master program.

3.6.1 Scan Complete

The Scan Complete lines are used to coordinate the action of the two instruments. This serves two functions, to coordinate the wavelength scans, and to insure that two motors are not unnecessarily turned on at the same time.

To coordinate wavelength scans, the master unit takes the following action.

Master Scan Complete	In	Out
.1 Wait for Slave unit	Ø	Ø
.2 Slave unit ready standby state	1	Ø
.3 Start Scan	1	1

A zero on the Master Scan Complete line means that the slave unit should complete its present action and then wait. A one means the slave unit can now start the next action.

The waits for the other instrument include a time-out so that a problem with one instrument does not disable the other.

3.6.2 Motor Delay

To insure that excessive current is not required to run all the motors in the entire system at once, the Master system waits until the slave system has had a chance to complete the following motor actions:

.1 Open Dust Cover
.2 Close Dust Cover
.3 Close Dark Shutter

When any of these actions are required, the master unit sets its Scan Complete Out line. This signals the slave unit to run its motor. When its action is complete, it sets its Scan Complete Out line. The master unit then resets its Scan Complete Out line to zero and runs its motor.

The slave unit takes no action while the platform latch is being withdrawn.

3.7 Slave Program Custom Points

The program for the slave instrument, FPS.SCR, has the following special features because it is the slave program.

3.7.1 Scan Complete

When the slave unit completes a wavelength scan it sets its Scan Complete Out high. This signals the Master unit. The slave unit then waits for its Scan Complete In to go high or for a time-out to run out. The Scan Complete In will indicate that Master Unit is ready for the slave unit to start its next scan.

3.7.2 Motor Start Delays

To insure that excessive current is not required to run all the motors in the entire system at once, the master unit waits until the slave system has had a chance to complete the following motor actions:

.1 Open Dust Cover
.2 Close Dust Cover
.3 Close Dark Shutter

When the slave unit receives a command requiring one of these actions, it sets its Scan Complete Out line low and waits for its Scan Complete In line to be set high. It then does the required action and sets its Scan Complete Out line high.

The slave unit takes no action while the platform latch is being withdrawn.

(End of Software Section)

SOFTWARE DEVELOPMENT SYSTEM

J. T. Riley
02/29/88
File: DEV612

INTRODUCTION

Procedures for down-loading the program from the development computer and for burning UV ROMs are also included.

4.0 Software Development System

The Software Development System is used for the following functions:

.1 Write flight software.
.2 Debug flight software.
.3 Exercise flight hardware and software.
.4 Change flight software between flights.
.5 Development Documentation.

It will perform the following activities.

.1 Write software
.2 Write documents.
.3 Store files on disk.
.4 Transmit software to flight microprocessor.
.5 Serve as flight microprocessor terminal during debugging.
.6 Receive the final flight software from the flight microprocessor.
.7 Burn the flight PROMs.

One Microprocessor Module is worked on at a time. The basic flight software is written on the Software Development System and transmitted over the RS-232 cable to a Microprocessor Module. There the program goes into temporary RAM. The software can then be tested one section at a time using the Development computer, as a terminal. All flight functions, except data transmision, may be proven on this system without the use of

Ground Support Equipment (GSE).

The flight program is maintained on disk in a separate IBM PC compatible microcomputer, the Software Development System. A Forth editor, or other word processing software, is used to enter and edit the flight program into the development computer and to save it on disk. Math and logic seccion of the flight program may be tested and debugged directly on the development microcomputer.

A utility program (DW) is then used to down-load the flight program into the Flight instrument's development RAM. After down-loading, the development computer becomes a terminal for the Flight instrument. The program can then be run and debugged. When problems are found, the flight instrument is stopped and the development editor run to fix the problem. This procedure allows the flight program to be tested in small blocks and problems are quickly eliminated.

When the program is complete and tested, it is burned into ROM. Two utility programs, one in the flight instrument (DUMP) and one in the development system (UP), work together to first set the Autostart and restart on error flags and then upload the application program from the development RAM to the development computer. The program is in the form of ASCII characters. The development utility stores the program on disk and retransmits it to the PROM burner.

The development RAM is then removed, the Flight/Dev jumper adjusted and the burned ROM installed in the same socket. The program can then be retested as an autostarting program in

ROM. The flight program may send status information out the RS-232 line to the development computer but it will ignore all incoming characters and cannot be interrupted from the keyboard.

To make interflight changes the same procedure is followed. The development RAM is re-installed and jumper set. The flight program is brought on the development computer's editor and any changes or additions are made. The program is then down-loaded and tested. When it is properly functioning it is burned into ROM and the new ROM installed in place of the old.

4.1 Hardware.

To develop the flight software and make changes between flights, a non-flight Software Development System is used (see Figure 20). This system consists of the following components:

.1 IBM Compatible PC
.2 Multipurpose card with RS-232 port.
.3 Power Supply
.4 Prom Burner
.5 Printer (optional)
.6 Development Cables
with Reset and RS-232 boxes.

4.1.1 Software Development Computer.

The Software Development Computer is an IBM compatible portable with a multipurpose card having a RS-232 and printer port, real time clock, and 360K of memory. See Figure 20, Software Development System.

A printer is helpful in software and document development but not mandatory and is not used in clean areas.

4.1.2 Power Supplies

The bench power supply provides enough +28 volt power to run bench tests for the entire instrument. Each instrument requires a little under 300 milliamps (TBD????) and the Platform motors require an additional 500 milliamp (TBD????).

A second small modular power supply is provided for the RS-232 line. Having this power supply separate allows start up tests to be run on the main supply.

The bench power supply is not provided under this contract, but the RS-232 supply is.

4.1.3 PROM Burner.

The PROM burner can be any available unit which can burn the type of ROMs used in the flight microprocessor cards and talk to the computer. It is only needed at the end of software development, after calibration, and after between flight software editing. It may either have its own card in the computer or communicate through the RS-232 port. Slight adjustments in software may be needed to suit a particular burner. Burners costing as little as $130 will serve this system well.

The PROM burner is not provided under this contract.

4.1.4 Cables

The cables connect the computer, power supplies, and flight instrument. The main communication cable may be up to 50 ft long if the flight instrument is to be maintained in a separate clean area.

The flight instrument end of the RS-232 cable terminates at the flight instrument end in a small box containing a +-12 Volt to +5 Volt signal level converter so that these components will not have to be flown. A second small box at the Development Computer end contains a Reset button and a jack for the RS-232 power supply.

Included in this contract are the long (50 ft) communications cable with two boxes and power supply, and a short (5 ft) power cable.

4.2 Set Up

Development or modification of the flight program requires:

.1 Communication Cable
with Reset Boxe and Power Supply
.2 Power Cable
.3 Bench Power Supply.
.4 PROM Burner
with software
.5 Software Development Computer
.6 Development Utilities
DW , UP

The development computer is connected to the flight instrument RS-232 cable. The power supply and Reset box on this cable independently power the RS-232 converter and allow the flight system to be reset from the Development Computer.

The RS-232 cable terminates in a small box which is in turn connected to the flight instrument by a short cable. This box contains a few components to convert the RS-232 +-12 volt signals to TTL levels and to optically isolate the two RS-232 signals. Since the power supply return in the Development Computer is connected to the green wire, safty ground, in its plug, the optical isollation is necesary to prevent a ground loop if the instrument supply is also case grounded at one point. Placing these components in a small development box allows them to be removed with the development cable and, thereby, not flown.

The RS-232 cable should be less than 50 feet, but can be long enough to allow the flight instrument to be maintained in a separate clean room.

4.3 Step By Step

The flight and ground programs may be edited and re-entered by the following step by step procedures.

4.4 Editing Programs

1. Make a Copy

First make a copy of the program to be edited under a new name. This may be done on the original or a new disk. The original program should not be changed directly but kept for reference. The copy may be made using the COPY command in MS-DOS :

```
A: COPY OLD-NAME.SCR B:NEW-NAME.SCR
```

Note that the extension .SCR is required by Forth and that on a new disk the Forth utility set having the extension .COM will be needed. You should also prepare a back-up disk for your program.

2. Boot Forth

Boot Forth with the new program.

```
FORTH   NEW-NAME
```

The Forth initiation display will appear on the monitor. Be sure that the program shown in all lower case letters is the program you wish to edit. If the name is not correct, then the correct file would not be found. Check your disk and try re-entering the name by USING NEW-NAME.

3. Enter Editor

```
EDIT
```

The editor display will appear on the monitor.

4. Set automatic Date

```
S
( Enter your Initials )
```

5. Edit a Screen

To edit a particular screen:

```
E
( Screen Number )
```

Check the editor section of the Forth documents to see the various editor commands available.

6. Leave Editor

To leave the editor press Esc twice. The edited program will automatically be written to disk.

7. Printout

To print out a copy of the entire new program:

```
Ø ?SCREENS SHOW
```

8. Leaving Forth

To leave Forth:

```
BYE
```

4.5 Down-Loading the Flight Program

1. Correct Program

Check that the program you wish to down-load is on the active disk and the program DW.COM is also on that disk.

2. Check Hardware

Check that the power is on to the flight microprocessor and that the development cable is attached. The Develpoment RAM must be installed and the memory switch on the development box in the RAM setting.

3. Reset Flight Microprocessor

Press the reset button on the RS-232 control box.

4. Set Baud Rate

Set RS-232 Baud rate. This must be redone after power up or after using any program which effects the COM1 command of the IBM (some word processors). The program MODE must be available on the disk in the active drive and may be done as part of the AUTO.EXE program.

```
A: MODE  COM1:48,N,7,2
```

5. Down-Load Program

Enter

```
DW.COM  NEW-NAME.SCR
```

Note that the extender .SCR must be entered. The monitor screen will clear and the message "Serial Transmission" appear.

6. Watch for Errors

The program is being transmitted and will appear on the screen without comments or blank lines. Watch for Error messages which may mean that there is a mistake in the program.

The Error messages are :

Dec Hex

Msg # 0 or	0	Word is not in Dictionary. Could be a bad number.
Msg # 1	1	Empty Stack. A Word expected something on stack.
Msg # 4	4	Isn't Unique. Not always a mistake.
Msg # 7	7	Parameter Stack full.
Msg # 12	C	Execution only.
Msg # 15	F	Protected Dictionary FORGET improperly used.
Msg # 17	11	Compilation only, use in definition.
Msg # 18	12	Declare Vocabulary FORGET improperly used.
Msg # 19	13	Conditional Not paired. Usually THEN missing.
Msg # 20	14	Definition Not Finished. Usually ; missing.

Other message numbers are not supported and should not appear.

Note the word in which the first error message occurs so that you can return to the editor and make the correction. If a word contains an error it will not be entered in the dictionary and all future references to it will cause additional errors. Usually you will be able to correct only the first one or two errors.

4.6 Burning the ROM

To burn a flight program into PROM, follow these steps:

1. Power up the Flight Instrument.

2. Load and test the flight program from development RAM.

3. Disk drive A must have a disk with the utility UP.COM on it and room to store the flight program in Hex.

4. The PROM burning program must be on the disk in drive A or drive B. This program is provided with the PROM Burner. The one used at RSI is a EPROM Writer Card made by ??????? and the program is called UPP512. The steps for using this system are

given below but may be slightly different for your system.

5. The Universal PROM Program socket box must be connected to the correct card in the Development Computer.

6. Up-load the program with the command:

UP.COM

Provide the file name with disk and extension when asked.

A:FILE-NAME.TSK

The file will be in executable code and stored in disk in ASCII.

7. Run the ROM burning program:

B:UPP15

Your system may have a different name for this program.

8. Select 27C64A at 12.5 volts.

E

5

9. Load program in memory

L
Starting Address = 0000
A:FILE-NAME.TSK

The extension .TSK is normal for a executable code file.

You may wish to review the file using the Display feature to be sure that it is not all zeros. In a good Forth program file you should see all but the last letter of each Forth word your defined.

10. Copy program to PROM

Place blank 27C64A PROM in correct socket. Besure that the #1 pin is in the correct location.

C

When the burning is complete the program will indicate this and that the verification is correct. If the PROM does not verify correctly, recheck all the steps particullary that the voltage is right for the chip and that the chip is in the correct socket.

11. Label PROM

Place a label on the PROM indicating the date and program. This lable should be readable with the #1 pin down and to the left.

The PROM is now ready to install in the board in place of the development RAM.

(End of Development Section)

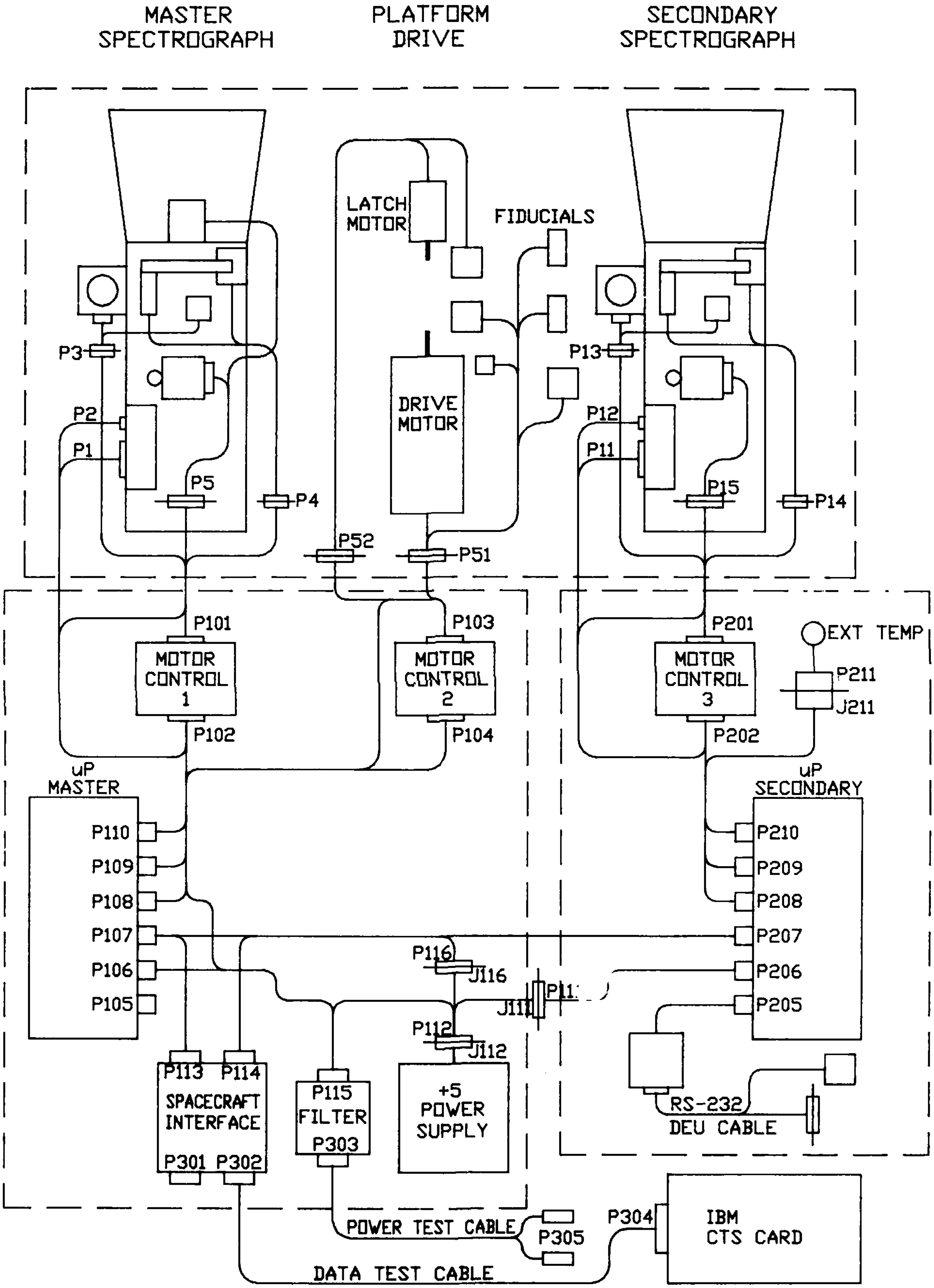
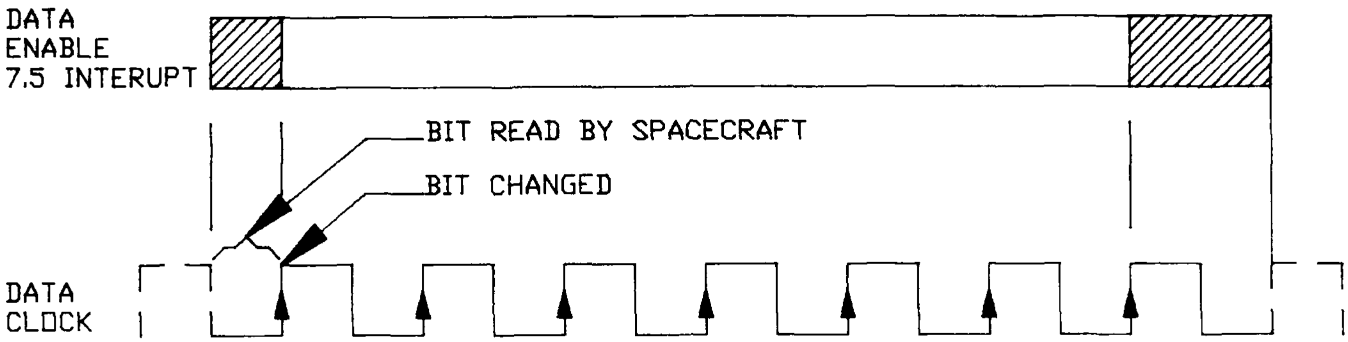


FIG 18

TEST CABLE BLOCK DIAGRAM

RILEY 11-17-88



1) ONE CLOCK LOW-HIGH FOR EACH BIT TRANSFERED
2) FIRST BIT ON LINE AT ENABLE
3) CLOCK RATE 10K< CLV <128K HZ
4) DUTY CYCLE FOR ALL DATA TRANSFER SHOULD BE LESS THAN 50%
5) ENABLE RATE 1 TO 100 HZ

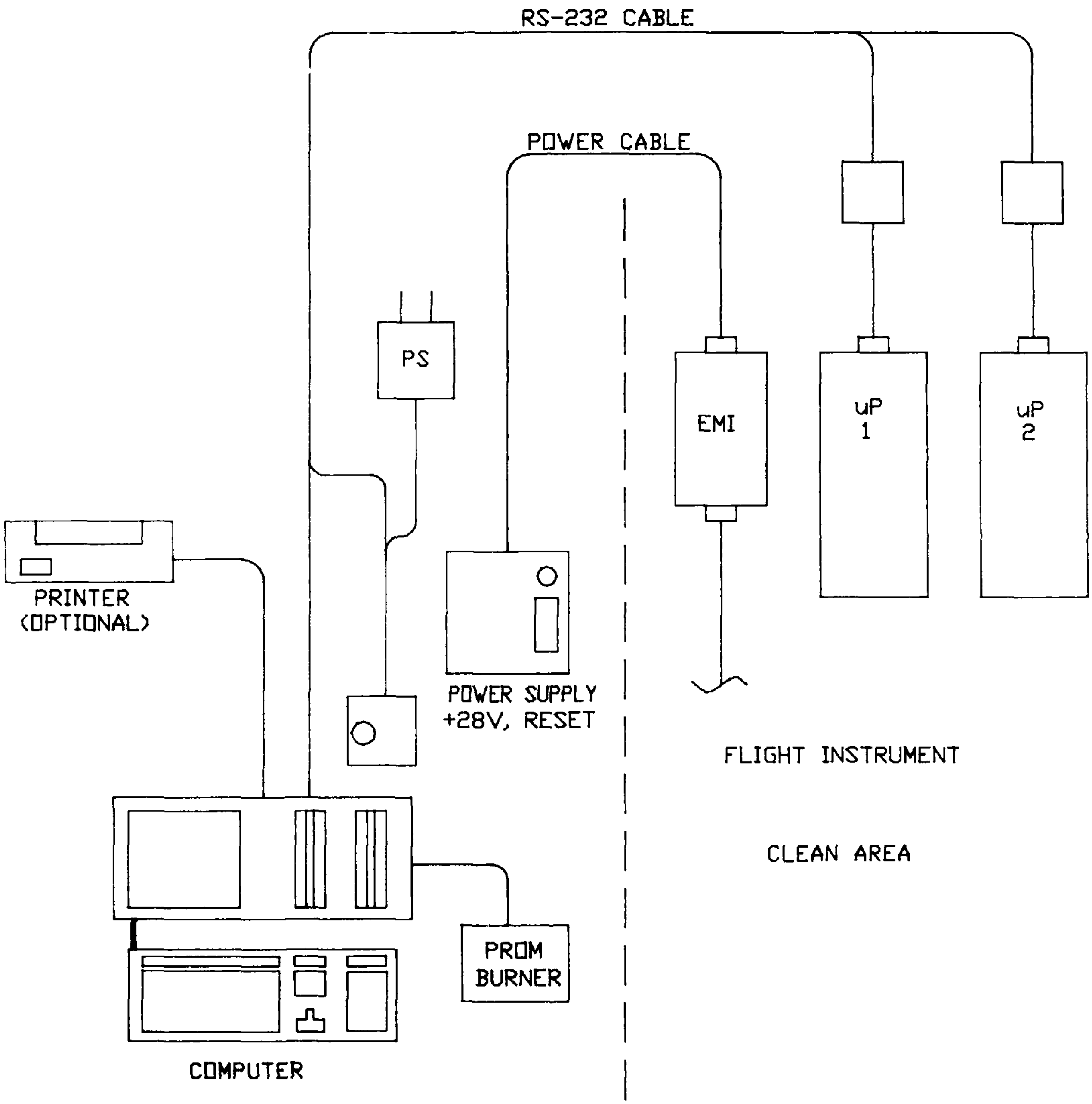
FIG 19 DATA CLOCKS

RILEY

7-17-87

FIG 20 SACS 2 INSTRUMENT

SOFTWARE DEVELOPMENT SYSTEM

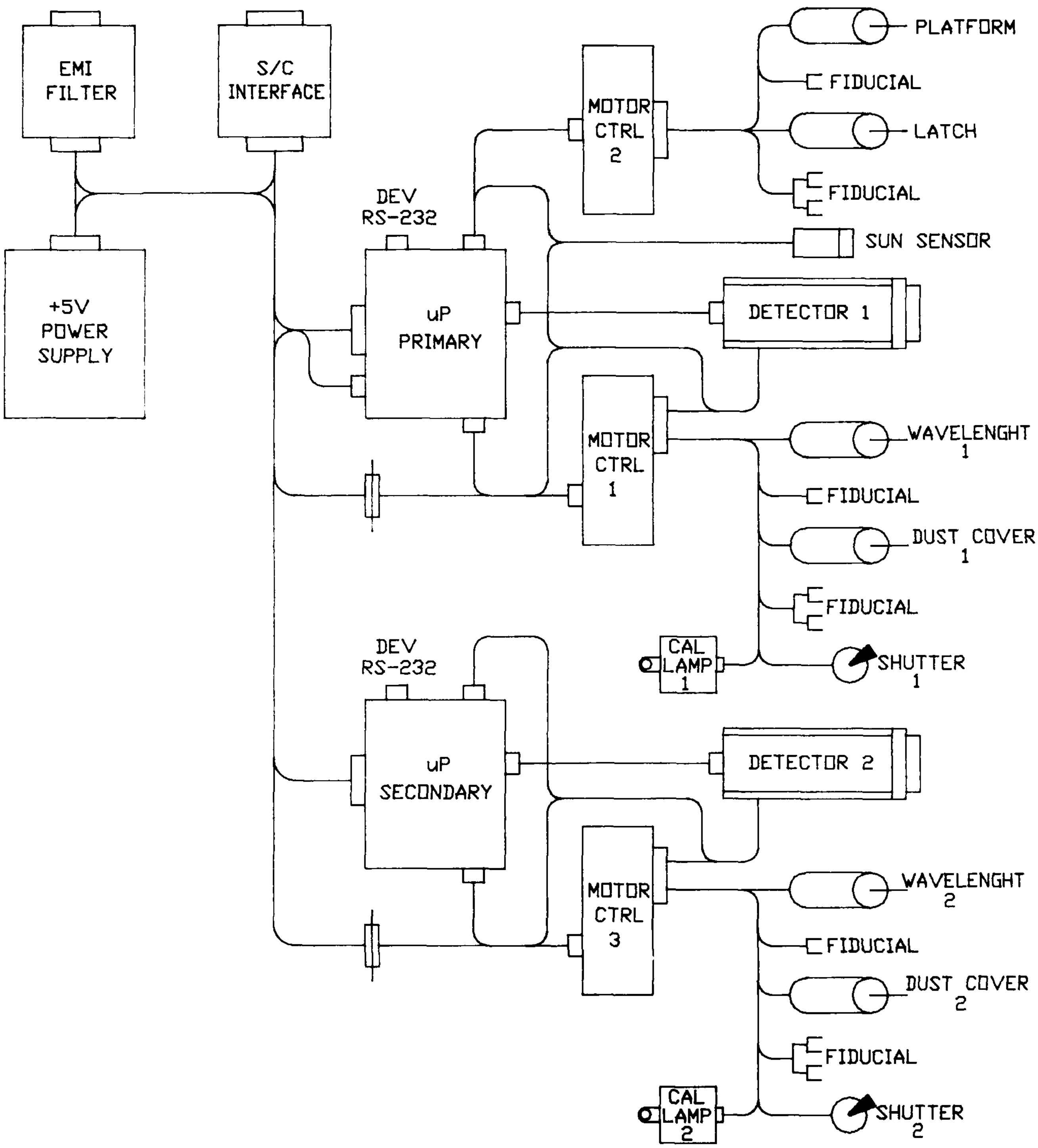


SOFTWARE DEVELOPMENT SYSTEM

RILEY
6-13-86

FIG 21

SACS 2 INSTRUMENT
MODULAR INSTRUMENT CONTROLLER



RILEY
6-12-86

Appendix IV

MEMORY AND I/O MAPS
FLIGHT INSTRUMENT
JOB 612 SACS

J. T. Riley
04/11/88

File: MAP612

INTRODUCTION

The memory and I/O are mapped for a 8085 Forth flight instrument.

1.0 Memory Map

1.1 Memory Block Types

The memory is made up of the following memory blocks:

All numbers are in Hex.

Start	End	Type	Chip	Description	Chip Select
0000	1FFF	PROM	27C64	Forth Core (8K)	0
2000	3FFF	PROM	27C64	Program (8K) (Note 1)	1 Flight
2000	3FFF	RAM	HM6264	Temporary RAM (8K) (Note 1) (remove before flight)	1 Dev
4000	50FF	RAM	8155H	Flight RAM in port (1/4K)	2
6000	60FF	RAM	8155H	Flight RAM in port (1/4K)	3
8000	80FF	RAM	8155H	Flight RAM in port (1/4K)	4
E000	E001	UART	8251A	UART for RS-232 line	7

Note 1. Either the PROM or RAM may be active during development but not both. The choice is controlled through a jumper on the board.

2.2 Assigned Memory Locations

Individual memory locations having specific designations defined by hardware or Forth requirements are:

Add	Mnem	Description
0000	RST0	Cold start or Reset vector (sent to CLD)
0008	RST1	NOT USED (sent to WRM)
0010	RST2	NOT USED (sent to WRM)
0018	RST3	NOT USED (sent to WRM)
0020	RST4	NOT USED (sent to WRM)
0024	TRAP	NOT USED (Sent to CLD)
0028	RST5	NOT USED (sent to WRM)
002C	RST5.5	Incoming RS-232 Character (sent to RST5.5)
0030	RST6	NOT USED (sent to WRM)
0034	RST6.5	NOT USED (sent to WRM)
0038	RST7	NOT USED (sent to WRM)
003C	RST7.5	Data Interrupt (sent to DATA vector)
0040		Start of Boot-up Literals (34 bytes)
0062		Start of Core and Assembly words (about 8K bytes)
	1FFF	End of Core space
2000		Vector to flight autostart Word (2 bytes)
2002		Vector to Data Out Word (3 bytes)
2006	2015	FORTH Vocabulary
2018		First Application Word (HERE at start of development)
	3FFF	End of Application Word area (about 8K bytes)
4000	4083	HERE in flight (154 bytes) Used by Word Buffer, and PAD.
4084	4085	User Area Pointer UP (2 bytes)
4086	40FF	User Variables (48 min, 100 bytes allotted)
6000	603F	Data Buffer (64 Bytes)
6040	60FF	Parameter Stack, S0 (170 min, 192 bytes allotted, grows down)
8000	803F	Temporary Input Buffer TIB (64 bytes)
8040	807F	KEYIN Buffer KIBUF (68 bytes)
8084	8085	Incoming Char Pointer ICHP (2 bytes)
8086	8087	Read Char Pointer RCHP (2 bytes)
8088	80FC	Return Stack, RP, IN (48 min, 116 bytes allotted, grows down)
80FE	80FF	Return Stack Pointer RPP (2 bytes)

```
E000            UART Data register  (1 byte)
E001            UART Control and Status Register (1 byte)
```

2.0 I/O Map

All chips are 8155H (port with RAM and timer/counter).

All numbers are in Hex.

```
I/O ADD Port   I/O     Description

PORT 2    CHIP  U10

41      A2 (8) O   Data Out (8)

42      B2 (8) I   Fiducials , or Analog In
                        1 - Wavelength Cam, course, Fid-0
                        2 - Wavelength Motor, fine, Fid-0
                        4 - Dust Cover Open, Fid-1
                        8 - Dust Cover Closed, Fid-1
                       10 - Scan Platform Cam, course, Fid-0
                       20 - Scan Platform Motor, fine, Fid-0
                       40 - Latch Open, Fid-1
                       80 - Latch Closed, Fid-1

                or
                   Analog In (4), Misc. (4)
                        1 - Sun Intensity Level 1, 1-High
                        2 - High Voltage Level 1, 1-High
                        4 - Temperature Level 1, 1-High
                        8 - One of above, Level 2, 1-High

                        10 - Optical Test Sensor, 1-On
                        20 - Output Data Counter, 1-MOD 8
                        40 - NOT Sun Present, 1-Normal
                        80 - Scan Complete Input, 1-Complete

43      C2 (6)  O  Miscellanous Output
                        1 - Data Out Buffer, Pulse to Preload
                            Data
                        2 - Fid / Analog Select  0-ANALOG/1-FID
                        4 - Scan Complete, 0-Normal /. 1-Scan
                            Complete
                        8 - Spare
                        10 - Spare
                        20 - Command Reset, 0-hold / 1-Reset

Timer/counter 2    Data Out counter (/8 Pulses)
```

PORT 3 CHIP U11

61	A3 (8) I	Data In Low Byte (8)
62	B3 (8) I	Data In High Byte (8)
63	C3 (6) O	1 - Data In Count Enable 0-Not/1-Enable 2 - Data In Counter Reset 0-Reset/1-Count Motor Control Lines (4), slow pulse 4 - Step Scan Motor 8 - Step Dust Cover Motor 10 - Step Platform Motor 20 - Step Latch Motor

Timer/counter 3 Not used

PORT 4 CHIP U9

81	A4 (8) I	Command In (8)
82	B4 (8) O	Relay Controls (8), 010 Pulse, 1 msec. 1 - Spectrometer Motor, Hold 2 - Spectrometer Motor, Step enable 4 - Dust Cover Motor, Hold 8 - Dust Cover Motor, Step enable 10 - Platform Motor, Hold 20 - Platform Motor, Step enable 40 - Latch Motor, Hold 80 - Latch Motor, Step enable
83	C4 (6) O	1 - Cal Lamp Relay, 0 - Off/ 1 - On 2 - Dark Shutter Optical Isolator, 0-OPEN/1-CLOSED 4 - Sun Inhibit Override 0-Inhibit enable/1-DET POWER ON 8 - Sun Not Present Reset (Pulse - reset) 10 - Spare 20 - Spare

Time/counter 4 BUAD Clock (16 * BAUD RATE)
1200 BAUD - 19200 Hz., Clk/ 160
4800 BAUD - 76800 Hz., Clk/ 40
9600 BAUD - 153600 Hz., Clk/ 20
Must be consistent with Core value.

3.0 USER Variables

Off Add	Description
00	Not Used
02	Not Used
04	Not Used

Start of Fixed Order Words

```
Ø6        SØ Address
Ø8        RØ Address
ØA        TIB Address
ØC        WIDTH (max) of word name
ØE        WARNING number of error
1Ø        FENCE
12        DP Address
14        VOC-LINK
                                   End of Fixed Order Words
16        BLK (Disk word not used)
18        IN  (Disk word not used)
1A        OUT (used by emit)
1C        SCR (disk word not used)
1H        OFFSET (disk word not used)
2Ø        CONTEXT vocabulary
22        CURRENT vocabulary
24        STATE compiling/executing
26        BASE of numbers printed
28        DPL output format
2A        FLD output format
2C        CSP error check
2E        R# (editing not used)
3Ø        HLD output format

                Start of Application USER Variables

32        First Application USER variable, see Screen # 1
.                 Allot 2 bytes per variable
.
.
64        Last available USER Variable
```

Appendix V

DATA BIT IDENTIFICATION

J. T. Riley
03/04/88

Introduction

The data from the instrument is in the form of a large serial data word. The size of the data word is defined for likely configurations and the meaning of each bit given.

1.0 Word Length

The length of the data contributed by each microprocessor module must be divisible by eight. The exact length of the word and the meaning for each bit in the word is dependent on the needs of an individual flight. This is set in software by DATA@ (screen #34) and by the constant DATA# (screen #2).

The first instrument, a single spectrometer with a scan platform, used the following bits:

Spectrometer	
Count	13
Command	3
Wavelength	10
Status	6
Angle	13
Angle Status	2
Error	1
Total	48

The proposed configurations would then have the following word lengths.

One Spec without Scan	
Count	16
Wavelength	10
Status	6
Command	4
Error	4

Total	40	5 Bytes
Two Spec. with Scan		
Master	40	
Scan	16	
Slave	40	
Total	96	12 Bytes
Four Spec. no Scan		
Master	40	
Slave 1	40	
Slave 2	40	
Slave 3	40	
Total	160	20 bytes

3.0 Data Timing

The microprocessor outputs data in response to a Data Enable and Data Clock (see Figure 19). The microprocessor expects these signals to be at TTL levels but they may be level shifted or inverted in the Spacecraft Interface module. Note that the Enable Signal must remain high during all the low to high transitions of the Data Clock. The Data Clock may be either gated or continuous. The microprocessor provides a new bit on each low to high transition of the clock and expects the spacecraft to read the bit just after each high to low.

The first bit of data will be available as soon as the Enable signal goes high, the Enable low to high transition must take place after a clock high to low. Similarly, the Enable signal must end after the last clock low to high and before the clock can go low to high again.

The Data Period, being the time between the start of data transfers, will be set by the spacecraft hardware and maybe any value between .01 and 1 seconds. If the data period is longer than the .01 second used on the first instrument, then additional Count bits will be needed (up to 16 bits) and the data word must then be extended. If the Data Period is longer than .1 second, the event counter may fill up and roll over on a bright source even with the full 16 bits.

The Data Clock rate is also set by the spacecraft and determines the time required to transfer the data. The total time require to transfer the data from all the instruments on a data channel should not exceed 70 of the Data Period or the system may not have time to execute commands properly. This rate may be any value between .5 KHz and 128 KHz. If the total data word is long, or the Data Period is short, the higher end of this range will be needed.

On multiple instrument configurations more than one data channel can be used to advantage. This will significantly improve data transfer rates. This feature would be implemented in hardware on the Spacecraft Interface card.

Software adjustment is necessary to set the Data Period for a mission (see DAT.EN.R screen 2), but the Data Clock rate does not affect software.

3.0 RS-232 Communication

A UART is included in the system to implement communication during development, but can be used for flight communication at lower rates than the primary data system. Baud rates of 1200, 4800, and 9600 are easy to establish by adjustment of the Baud rate number in the Forth Core, but reburning of the core is required. A rate of 19.2K Baud is the fastest possible and 200 Baud is the slowest. Receiving programs for the IBM PC writen in PC/Forth will have trouble reading at Baud rates above 4800. An AT or a assembly language program can run at the higher rates.

The input and output signals are at TTL levels of 0 and +5 Volts and must be shifted to either the -12 Volt / + 12 Volt levels for RS-232 or the complementary 0 / + 5 Volt levels for RS-422. During development a small adaptor box is provided in the Development cable for this shift and for optical isolation. This box is powered by an external power supply to allow testing of the Autostart routine with the communication link already operational. If this communication link is to be used in flight, appropriate level shifters will be needed on the Spacecraft Communication Card.

It is possible to modify the software so that flight commands are received through this communication channel. It is even possible to add additional Forth words to the program in flight.

If this channel is not used in flight then the UART chip (8251A) may be removed to save power.

4.0 Bit Identification

The meaning of each bit of the Data can be set in software. The total length of Data for each microprocessor must be divisible by eight to comply with standard word length even if unchanging filler bits or extra error bits have to be added. The number of Count Bits and the presents of a scan platform affect the total number of data bits directly.

The following example shows three (3) configurations. The first has a data bit pattern identical with the first engineering instrument, the second is a single instrument with no platform, and the third has two instruments and a platform.

4.1 Spectrometer with Platform.

First Byte	
1	Count, low word, bit 1, LSB
2	Count, low word, bit 2
3	Count, low word, bit 3
4	Count, low word, bit 4
5	Count, low word, bit 5
6	Count, low word, bit 6
7	Count, low word, bit 7
8	Count, low word, bit 8
Second Byte	
9	Count, high word, bit 9
10	Count, high word, bit 10
11	Count, high word, bit 11
12	Count, high word, bit 12
13	Count, high word, bit 13, MSB
14	Command word, bit 1
15	Command word, bit 2
16	Command word, bit 3

Third Byte

17	Wavelength, low word, bit 1, LSB
18	Wavelength, low word, bit 2
19	Wavelength, low word, bit 3
20	Wavelength, low word, bit 4
21	Wavelength, low word, bit 5
22	Wavelength, low word, bit 6
23	Wavelength, low word, bit 7
24	Wavelength, low word, bit 8

Fourth Byte

25	Wavelength, high word, bit 9	
26	Wavelength, high word, bit 10, MSB	
27	Wavelength Fiducial.	1 = At fid
28	Cover Status,	1 = Closed
29	Shutter Status,	1 = Closed
30	Test Lamp Status,	1 = On
31	High Voltage Status,	1 = On
32	Solar Sensor Status	1 = Sun Present

Fifth byte

33	Angle, low word, bit 1, LSB
34	Angle, low word, bit 2
35	Angle, low word, bit 3
36	Angle, low word, bit 4
37	Angle, low word, bit 5
38	Angle, low word, bit 6
39	Angle, low word, bit 7
40	Angle, low word, bit 8

Sixth Byte

41	Angle, high word, bit 9	
42	Angle, high word, bit 10	
43	Angle, high word, bit 11	
44	Angle, high word, bit 12	
45	Angle, high word, bit 13, MSB	
46	Scan Angle Fiducial,	1 = At fid
47	Latch Status,	1 = Latched
48	Error Bit,	1 = Error.

This is the data bit definition used in the first instrument.

4.2 Spectrometer Alone

First Byte	
1	Count, low word, bit 1, LSB
2	Count, low word, bit 2
3	Count, low word, bit 3
4	Count, low word, bit 4
5	Count, low word, bit 5
6	Count, low word, bit 6
7	Count, low word, bit 7
8	Count, low word, bit 8
Second Byte	
9	Count, high word, bit 9
10	Count, high word, bit 10
11	Count, high word, bit 11
12	Count, high word, bit 12
13	Count, high word, bit 13,
14	Count, high word, bit 14
15	Count, high word, bit 15
16	Count, high word, bit 16, MSB
Third Byte	
17	Wavelength, low word, bit 1, LSB
18	Wavelength, low word, bit 2
19	Wavelength, low word, bit 3
20	Wavelength, low word, bit 4
21	Wavelength, low word, bit 5
22	Wavelength, low word, bit 6
23	Wavelength, low word, bit 7
24	Wavelength, low word, bit 8
Fourth Byte	
25	Wavelength, high word, bit 9
26	Wavelength, high word, bit 10, MSB
27	Wavelength Fiducial. 1 = At fid
28	Cover Status, 1 = Closed
29	Shutter Status, 1 = Closed
30	Solar Sensor Status 1 = Sun Present
31	High Voltage Status, 1 = On
32	Test Lamp Status, 1 = On

Fifth Byte

33	Command word, bit 1
34	Command word, bit 2
35	Command word, bit 3
36	Command word, bit 4
37	Error Bit
38	Error Bit
39	Error Bit
40	Error Bit

Compared to the first instrument, the count has been extended to 16 bits to allow longer integration times. The Command Word has been moved to the last byte and extended by one bit so that it can show the exact command number instead of the command number minus one.

4.2 Dual Spectrometers

Dual spectrometers with a scaning platform controlled by the master unit.

First Byte

1	Count, low word, bit 1, LSB
2	Count, low word, bit 2
3	Count, low word, bit 3
4	Count, low word, bit 4
5	Count, low word, bit 5
6	Count, low word, bit 6
7	Count, low word, bit 7
8	Count, low word, bit 8

Second Byte

9	Count, high word, bit 9
10	Count, high word, bit 10
11	Count, high word, bit 11
12	Count, high word, bit 12
13	Count, high word, bit 13,
14	Count, high word, bit 14,
15	Count, high word, bit 15,
16	Count, high word, bit 16, MSB

Third Byte

17	Wavelength, low word, bit 1, LSB
18	Wavelength, low word, bit 2
19	Wavelength, low word, bit 3
20	Wavelength, low word, bit 4
21	Wavelength, low word, bit 5
22	Wavelength, low word, bit 6
23	Wavelength, low word, bit 7
24	Wavelength, low word, bit 8

Fourth Byte

25	Wavelength, high word, bit 9	
26	Wavelength, high word, bit 10, MSB	
27	Wavelength Fiducial.	1 = At fid
28	Cover Status,	1 = Closed
29	Shutter Status,	1 = Closed
30	Solar Sensor Status	1 = Sun Present
31	High Voltage Status,	1 = On
32	Test Lamp Status,	1 = On

Fifth Byte

33	Command word, bit 1
34	Command word, bit 2
35	Command word, bit 3
36	Command word, bit 4
37	Error Bit
38	Error Bit
39	Error Bit
40	Error Bit

Sixth byte

33	Angle, low word, bit 1, LSB
34	Angle, low word, bit 2
35	Angle, low word, bit 3
36	Angle, low word, bit 4
37	Angle, low word, bit 5
38	Angle, low word, bit 6
39	Angle, low word, bit 7
40	Angle, low word, bit 8

Seventh Byte

41	Angle, high word, bit 9	
42	Angle, high word, bit 10	
43	Angle, high word, bit 11	
44	Angle, high word, bit 12	
45	Angle, high word, bit 13, MSB	
46	Scan Angle Fiducial,	1 = At fid
47	Latch Status,	1 = Latched
48	Error Bit,	1 = Error.

Eight Byte		Start of Slave Instrument
1	Count, low word, bit 1, LSB	
2	Count, low word, bit 2	
3	Count, low word, bit 3	
4	Count, low word, bit 4	
5	Count, low word, bit 5	
6	Count, low word, bit 6	
7	Count, low word, bit 7	
8	Count, low word, bit 8	
Ninth Byte		
9	Count, high word, bit 9	
10	Count, high word, bit 10	
11	Count, high word, bit 11	
12	Count, high word, bit 12	
13	Count, high word, bit 13	
14	Count, high word, bit 14	
15	Count, high word, bit 15	
16	Count, high word, bit 16, MSB	
Tenth Byte		
17	Wavelength, low word, bit 1, LSB	
18	Wavelength, low word, bit 2	
19	Wavelength, low word, bit 3	
20	Wavelength, low word, bit 4	
21	Wavelength, low word, bit 5	
22	Wavelength, low word, bit 6	
23	Wavelength, low word, bit 7	
24	Wavelength, low word, bit 8	
Eleventh Byte		
25	Wavelength, high word, bit 9	
26	Wavelength, high word, bit 10, MSB	
27	Wavelength Fiducial.	1 = At fid
28	Cover Status,	1 = Closed
29	Shutter Status,	1 = Closed
30	Solar Sensor Status	1 = Sun Present
31	High Voltage Status,	1 = On
32	Test Lamp Status,	1 = On
Twelveth Byte		
33	Command word, bit 1	
34	Command word, bit 2	
35	Command word, bit 3	
36	Command word, bit 4	
37	Error Bit	
38	Error Bit	
39	Error Bit	
40	Error Bit	

The count is extended to 16 bits to allow longer integration times. The Command Word is moved to the last byte and extended by one bit so that it can show the exact command number instead of the command number minus one.

The data bit identifications for the two (2) instruments are identical, but two bytes of platform data have been added to the master instrument.

Appendix VI

FORTH CORE VOÇABULARY
MICROPROCESSOR 8Ø85
8K CORE

J. T. Riley
Ø2/11/87
FILE: WORDS85

Introduction

The WORDs in the FORTH core are listed and briefly described.

1.Ø FORTH

The following words are included in the FORTH Vocabulary. Flight Words will be added to this vocabulary.

The abbreviation in the stack explanation have the following meanings:

N N1 ...	16 bit number
D D1 ...	32 bit number
U U1 ...	16 bit unsigned number
addr	memory address
byte	8 bit byte
char	8 bit ASCII character
F	Bollean Flag: Ø=False, any number=True but -1 preferred
rem quot ...	16 bit numbers
name	Naming string

1.1 Stack

Word	Stack Effect	Function
DUP	N1 -- N1 N1	Duplicate top number
2DUP	D1 -- D1 D1	Duplicate Double number
-DUP	Ø -- Ø OR N1 -- N1 N1	Dup if not zero
SWAP	N1 N2 -- N2 N1	Swap top two numbers
2SWAP	D1 D2 -- D2 D1	Swap doubles
ROT	N1 N2 N3 -- N2 N3 N1	Bring up 3rd number
OVER	N1 N2 -- N1 N2 N1	Copy 2nd to top
DROP	N --	Discard top number
R	N --	Move N to Return Stack
R	-- N	Pop the Return Stack

1.2 Conversion

HEX	--	Set BASE to 16
DECIMAL	--	Set BASE to 1Ø

The following number formating words are available but not normally used by flight programs:

DIGIT # # SIGN # #S HOLD

1.3 Arithmetic

+	N1 N2 -- N3	Addition
D+	D1 D2 -- D3	Double addition
-	N1 N2 -- N3	Subtract
D-	D1 D2 -- D3	Double Subtraction
*	N1 N2 -- N3	Multiply
U*	U1 U2 -- U3	Unsigned multiply
M*	N1 N2 -- D	Multiply with double result
/	N1 N2 -- quot	Divide
/MOD	N1 N2 -- rem quot	Divide leaves rem and quot
MOD	N1 N2 -- rem	Modulo, remainder
U/	U1 U2 -- U3	Unsigned divide
M/	D N1 -- rem quot	Divide leaves rem and quot
M/MOD	D N -- rem quot	Mixed precision divide
*/	N1 N2 N3 -- N4	N4 = (N1*N2)/N3
*/MOD	N1 N2 N3 -- rem quot	*/ with remainder
M*/	D1 N1 N2 -- D2	Mixed */
Ø	-- Ø	Place Ø on stack
1	-- 1	Place 1 on stack
2	-- 2	Place 2 on stack
3	-- 3	Place 3 on stack
4	-- 4	Place 4 on stack
5	-- 5	Place 5 on stack
6	-- 6	Place 6 on stack
7	-- 7	Place 7 on stack
8	-- 8	Place 8 on stack
9	-- 9	Place 9 on stack
1+	N1 -- N2	Increment by 1
2+	N1 -- N2	Increment by 2
1-	N1 -- N2	Decrement by 1
2-	N1 -- N2	Decrement by 2
2*	N1 -- N2	Multiply by 2 fast
4*	N1 -- N2	Multiply by 4 fast
8*	N1 -- N2	Times 8 , high speed
ABS	N1 -- N2	Absolute value
DABS	D1 -- D2	Double Absolute
MINUS	N1 -- N2	Change sign
DMINUS	D1 -- D2	Change sign double
S-D	N -- D	Extend sign to double
+-	N1 N2 -- N3	Apply sign N1 to N2
D+-	D1 D2 -- D3	Apply sign D1 to D2
MIN	N1 N2 -- N3	Leave larger number
MAX	N1 N2 -- N3	Leave smaller number

1.4 Logical

AND	N1 N2 -- N3	Bitwise logical AND
OR	N1 N2 -- N3	Bitwise logical OR
XOR	N1 N2 -- N3	Bitwise exclusive-OR
1COM	N1 -- N2	One's Complement

1.5 Comparison

0=	N1 -- F	True if N1 = 0
0	N1 -- F	True if N1 0
=	N1 N2 -- F	True if N1 = N2
	N1 N2 -- F	True if N1 N2
U	U1 U2 -- F	True if U1 U2
	N1 N2 -- F	True if N1 N2
NOT	F1 -- F2	Toggle Flag
D0=	D1 -- F	True if D1 = 0
D=	D1 D2 -- F	True if D1 = D2
D	D1 D2 -- F	True if D1 D2
D	D1 D2 -- F	True if D1 D2

All flags are 0 False or -1 True.

1.6 Memory

@	addr -- N	Fetch single number
C@	addr -- byte	Fetch single byte
!	N addr --	Store N at address
C!	byte addr --	Store byte at address
CMOVE	addr1 addr2 len --	Move Byte string lo-hi
+!	N addr --	Add N to number at address
TOGGLE	addr byte --	XOR byte at address
FILL	addr N byte --	Fill memory and byte
BLANKS	addr count --	Fill With blanks
D@	addr -- D	Fetch Double Number
D!	D addr --	Store Double Number

1.7 Input / Output

KEY	-- char	Input character from RS-232
EXPECT	addr count --	Console string input
EMIT	char --	Output character to RS-232
TYPE	addr len --	Output character string
CR	--	Output CR LF to RS-232
."	--	Output string to "
D.R	D fieldwidth --	Output right justified
D.	D --	Output double number RS-232
.	N --	Output signed number
B.	byte --	Output byte
SPACE	--	Output one space
SPACES	N --	Output N spaces

1.8 Strings

The following string words are rarely used in flight programs.

-TRAILING WORD COUNT NUMBER

1.9 System

The following system Words are rarely used in flight programs.

```
LIT      EXECUTE SP@     SP!     RP!     +ORIGIN S0
R0       TIB     WIDTH   WARNING FENCE   DP      VOC-LINK
IN       OUT     CONTEXT CURRENT STATE   BASE    DPL
FLD      CSP     HLD     HERE    LFA     CFA     NFA
PFA      !CSP    PAD     ERROR   ID.     INTERPERT
IMMEDIATE        VOCABULARY      QUIT    ABORT   WARM
COLD     ?ERROR  ?COMP   ?EXEC   ?PAIRS  ?CSP    -FIND
?STACK   BYE     MESS    BACK    LATEST  QUERY   COMPILE
TRAVERSE
```

1.10 CONTROL

```
IF ...        F  --               Conditional execution
  ELSE ...                          ( optional )
  THEN                              continue

I                     --  N       DO loop index
DO ...   limit index --           Interative Loop
  LOOP
DO ...   limit index --           Do loop with specified
  +LOOP        N      --            increment N
LEAVE                 --          Jump out of Do loop

BEGIN ...                         Loop until flag is true
  UNTIL        F      --
BEGIN ...                         Loop while flag is true
  WHILE ...    F      --
  REPEAT
BEGIN ...                         Loop forever
  AGAIN
```

1.11 Defining Words

```
FORTH                             Makes FORTH vocabulary CONTEXT
ASSEMBLER                         Makes ASSEMBLER CONTEXT
CONSTANT name  value --         Define constant
VARIABLE name      --           Create variable
                                    ROM in flight
USER   name   offset --         Create user variable
                                    RAM in flight
ALLOT          N      --          Allot memory space after HERE

: name               -- addr    Start colon definition
;                                 End colon definition
,              N      --          Comma, store N at HERE
CREATE name                     Build dictionary header
' name               -- addr    Tick, leave PFA
SMUDGE                            Toggle smudge bit
;CODE                             Switch to Assembly in
                                     middle of colon dif.
[COMPILE] name                  Force compilation of an
                                     immediate definition
[                                 Sets Interpret State
```

	]		Set Compilation State
	BUILDS		Used in defining Words
	DOES		Used in defining Words
	LITERIAL	N --	Compile N as literal
	DLITERIAL	D --	Compile D as literal

1.12 UTILITIES

	FORGET name		Forget Words starting with name
	(		Start comment until)
	?	addr --	Output address contents
	NOOP	--	No operation
	VLIST	--	List words in dictionary
	DEPTH	-- N	Number of numbers on stack
	.S	--	Non-destructive stack list
*	EI5.5	--	Enable 5.5 Interrupt
*	EI7.5	--	Enable 7.5 Interrupt.
*	EI7.55.5	--	Enable 7.5 and 5.5
*	AUTO	--	Run autostart word.
*	DUMP	--	Output entire flight program (in Executable code.)
*	DUMPHEX	--	Output entire flight program (in Intel Hex format.)
	ON	-- -1	Put true flag on stack
	OFF	-- Ø	Put false flag on stack
*	OPEN	-- Ø	Put false flag on stack
*	CLOSE	-- -1	Put true flag on stack
*	WT	N --	Short delay
*	MSEC	N --	Delay in milliseconds
*	IRET	--	Return from Forth interrupt
*	DIS.I	--	Disable all interrupts
*	BEEP	--	Output ASCII 7 , bell
*	FLAG	N -- F	Ø or -1
*	--	--	Next Page, NOOP

* Not standard Fig Forth

Appendix VII

FORTH ASSEMBLER

8085

JOB 972 SSBUV

J. T. Riley
07/15/87

Introduction

The procedure for writing Forth Code (assembly) words is detailed for the 8085.

1.0 Code Words

To increase execution speed, Forth words may be written in assembly language. The definitions of these words starts with CODE and ends with C; and are written with the help of a small 8085 assembler included in Utility section of the core.

Since this assembler could not follow both Forth syntax and Intel's 8085 syntax, its structure is a bit of a compromise. Also, available memory space is limited so not all 8085 instructions are supported but, if necessary, it is possible to add most of the missing ones by simply adding the required word to the Forth vocabulary.

A typical 8085 assembly word would look like this:

```
ASSEMBLER  HEX
CODE  P@.25  (  -- N, Read Port 25H )
        25  IN          ( Read port )
        A   L   MOV     ( Move to L )
        0   H   MVI     ( Zero high byte )
        HPUSH   JMP  C; ( Put on stack )

FORTH
```

Note that the parameters, registers and data, come before the instruction as is normal for Forth. The first parameter is the source and the second is the destination which is the opposite of normal assembly language for the 8Ø85. The pnemonics are, however, standard for the 8Ø85.

If you are using a Forth editor with screens, Code word definitions may be continued from one screen to the next with only a -- (Next Page). **Colon definitions must be completed in the same screen.**

2.Ø Pnemonics Supported

The following 8Ø85 pnemonics are supported:

8Ø85		Description
Registers		
A	A	A Register, 8 Bits or PSW and A register, 16 Bits
B	B	B Register, 8 Bits or BC register pair, 16 bits
C	C	C Register, 8 Bits
D	D	D Register, 8 Bits or DE register pair, 16 Bits
E	E	E Register, 8 Bits
H	H	H Register, 8 Bits or HL register pair, 16 Bits
L	L	L Register, 8 Bits
M	M	Memory addressed by HL contents.
SP	SP	Stack Pointer Register, 16 Bits
Address Constants		
	NEXT	Jump for next Forth word
	HPUSH	Jump for next Forth word and put HL contents on Stack

One Instruction Byte, no registers 1MI

RAL	RAL	Rotate Left Accumulator
RAR	RAR	Rotate Right Accumulator
CMA	CMA	Complement the Accumulator
EI	EI	Enable Interrupts
DI	DI	Disable Interrupts
RET	RET	Return from subroutine
PCHL	PCHL	Jump to the memory location in HL
XCHG	XCHG	Exchange the contents of DE and HL
RIM	RIM	Read Interrupt Mask
SIM	SIM	Set Interrupt Mask

One Instruction Byte, 1* Spacing of Registers 2MI

ADD reg	reg ADD	Add 8 bit register to the A register
SUB reg	reg SUB	Sub 8 bit register from A register
ANA reg	reg ANA	And 8 bit data against A register
ORA reg	reg ORA	And 8 Bit Data with A register
CMP reg	reg CMP	Compare register with A register

One Instruction Byte, 8* Spacing of Registers 3MI

INR reg	reg INR	Increment 8 bit Register
DCR reg	reg DCR	Decrement 8 bit Register
PUSH reg	reg PUSH	Push register pair on return stack
POP reg	reg POP	Pop register pair from return stack
INX rp	rp INX	Increment 16 bit Register pair
DCX rp	rp DCX	Decement 16 Bit Register Pair

One Instruction Byte, One Data Byte 4MI

ADI A,data	data ADI	Add 8 bit data to A register immediate
SUI data	data SUI	Sub 8 bit data from A regiscer immediate
ANI data	data ANI	AND 8 bit data with A register immediate
ORI data	data ORI	OR 8 bit data with A register immediate
XRI data	data XRI	XOR 8 bit data with A register immediate
IN port	port IN	Input 8 bits into the A register
OUT port	port OUT	Output 8 bits from A register

One Instruction Byte, Two Data or Register Bytes 5MI

JMP addr	addr JMP	Jump to address
STA addr	addr STA	Store A register in address
LDA addr	addr LDA	Load contents of address in A register
SHLD addr	addr SHLD	Load contents of HL into address
LHLD addr	addr LHLD	Load address contents into HL reg.
MOV reg2,regl	regl reg2 MOV	* Move 8 bits between registers or to memory indicated in HL
MVI reg,data	data reg MVI	* Load 8 bits of data in register.
LXI rp,data	data rp LXI	* Load 16 bit data into register pair.

* Note order of registers, from the first entry to the second.

Where:

data	An Integer number
reg	8 bit register, (A B C D E H L or M) or
rp	Register pair, 16 Bits, (SP A B C H)
addr	Address or constant used as an address
port	Port address

Missing pnemonics can only be added to the Forth vocabulary. The ASSEMBLER vocabulary is contained completely in PROM so its length cannot be changed. The Forth names of the pnemonics must be different from all other words defined to that point; words in the ASSEMBLER vocabulary proper can have the same name as a different word in the Forth vocabulary (IF, ELSE, etc.).

Many of the missing pnemonics can be defined using the defining words 1MI through 5MI shown in the group headings. The correct group for a pneumonic can be determined from the reference below or by finding existing pnemonics with similar features. The pneumonic's numeric equivalent is also needed.

The BC and SP registers must be preserved across all Forth words. The words PUSH and POP may be used to get numbers on and off the parameter stack and need not be used in pairs. The HL register is normally used to place one parameter on the stack at the end of a word.

3.0 Conditional Control

Complicated control structures are best handled as several Forth words, but simple conditional jumps may be included within an assembly word. The assembler's handling of conditional jumps is closer to Forth than to a normal 8085 assembler. The IF-ELSE-ENDIF and BEGIN-UNTIL control structures are supported by conditional words.

The following words are included:

```
Conditionals                                      ( Compiles )
        0=      True if equal to Zero             ( JNZ )
        0       True if greater then Zero         ( JP )
        NOT     Reverses above conditionals.      ( JZ or JM )

Control Structures
        IF      Start IF-ENDIF  *
        ELSE    Alternative in IF-ELSE-ENDIF*
        ENDIF   End of conditional

        BEGIN   Start of BEGIN-UNTIL *
        UNTIL   end of BEGIN-UNTIL *
                * Not the same word as in the FORTH Vocabulary.
```

Labeling for these control structures is handled automatically by the assembler.

The following word reads an I/O location and outputs a flag to indicate if the reading is above or below a value held in a constant (REF.VAL).

```
ASSEMBLER  HEX
CODE    P@25.F    (  --   f , Read Port 25 and set Flag )
        25  IN              ( Read port )
        REF.VAL  H LXI      ( Get reference Value)
        L   CMP             ( Set flags )
     0=   IF                ( Start conditional)
          -1   H  LXI       ( On flag )
     ELSE
          00   H  LXI       ( Off flag )
     ENDIF
        HPUSH JMP  C;
FORTH
```

Appendix VIII

FORTH REFERENCE

J. T. Riley
04/11/88

Introduction

The following reference materials were used to establish the RSI Forth flight system and are critical to understanding it.

1.0 Forth Language

The best reference for the general use of the Forth Language is:

Thinking Forth
Leo Brodie
Prentice-Hall
Englewood Clifs, New Jersey 07632

This book does, however, use PolyForth from Forth Incorporated and not FigForth, there are some differences, and it avoids machine specific topics such as interrupts.

2.0 FigForth Core

The Forth core used in RSI systems comes from:

FigForth For 8085
Assembly Source Listing
Release 1.1
Forth Interest Group
P.O. Box 1105
San Carlos, CA 94070

This and all other Forth materials listed herein were obtained through Mountain View Press below.

3.0 FigForth Core Operation

A detailed explanation of the working of the FigForth core can be found in:

System Guide to FigForth
C. H. Ting
Mountain View Press Inc., 1982
(415) 961-4103

Additional information on Forth System operation, such as standard error messages, can be found in:

Forth Fundamentals Vol 1
C. Kevin McCabe
Dilithium Press
Beaverton, Oregon 1983

4.0 FigForth Vocabulary

A detailed explanation of Forth words can be found in the following references

All About Forth
An Annotated Glossary
Glen B. Haydon
Mountain View Press Inc., 1984

5.0 Assembler

The assembler is a modification of the 8085 Assembler from Chapter 14 of "Systems Guide to FigForth" by C. H.Ting (3.0 above).

6.0 IBM Forth

The Forth programs which are run on the IBM PC, (DW , UP) are written in:

PC/FORTH
Laboratory Microsystems, Inc.
P.O. Box 10430
Marina del Rey, CA 90295

Appendix IX

DUAL SPECTROMETER WITH PLATFORM
JOB 612 HUP2

J. T. Riley
11/23/88

File: DUAL

INTRODUCTION

The specific execusion of the dual spectrometer with platform are detailed.

1.0 Dual Spectrometer System

The first instrument to use the Forth microprocessors will be a two spectrometer instrument with scan platform. This instrument does not unfortunately presently manifested for launch. It is necessary to build up those portions of the system which will have to be altered to suit the spacecraft as handbuilt test units. Flight versions will be built later to suit the mission.

1.1 Hardware Construction Level

The following pieces are being built and cabled for flight:

1. Master Spectrometer
2. Secondary Spectrometer
3. Master Microprocessor
4. Secondary Microprocessor
5. Platform cabling

The following pieces are laboratory operation only:

1. System Cabling
2. Spacecraft Interface Box
3. Input Power Filter Box
4. +5 Power Supply

A problem was discovered with the following piece during development and they were modified. These modified units may not be acceptable for flight.

1. Motor Control Box #1
2. Motor Control Box #2
3. Motor Control Box #3

The following components are used for development and testing. They are not flown.

1. Data Test Cable
2. Power Test Cable
3. RS-232 Development Cable
4. IBM CTS Card

1.2 Analog Signals

The following analog signals occur in the dual spectrometer with platform system:

Signal	Micro	Spacecraft
1. Det #1 HV Monitor	Master (2)	Yes
2. Det #1 Temperature	no	Yes
3. Sun Sensor #1 Intensity	no	Yes
4. Det #2 HV Monitor	Secondary (2)	Yes
5. Det #2 Temperature	Secondary (1)	Yes
6. Plat Pot	no	Yes
7. System Temperature	no	Yes
8. Plat Fid position 1	Master (1)	no
9. Plat Fid position 2	Master (1)	no
10. Spare	Secondary (1)	no

The 8 and 9 use extra analog inputs to supplement the fiducial readers.

2.0 Software

The flight programs PRM.SCR and FPS.SCR are used for this system.

Appendix X

MODIFICATIONS BEFORE FLIGHT
FLIGHT INSTRUMENT
JOB 612 SACS

J. T. Riley
Ø1/Ø5/89

File: MOD612.DOC

INTRODUCTION

The modifications and completions to the flight instrument which will be needed when a flight is defined are laid out.

1.Ø Hardware Flight Tailoring

When a specific flight is defined the following considerations will affect hardware in the system:

1. Presence of scan platform.
2. Number of instruments in system (1 to 4)
3. Type of Spacecraft Interface.
4. Physical layout of components.

This will affect the following components:

1. Spacecraft Interface Box
2. System Cabling
3. Platform motor driver.
4. Power Filter Box.

Adaptation to most rocket and short term shuttle missions should not require redesign of other components.

The Spacecraft Interface Box will have to be adapted to the specific spacecraft. This might mean adding optical isolators or shifting signal levels. It is also possible to use the RS-232 line for spacecraft communication, in which case level shifters and line drivers will be needed as well as software alterations.

An example of the system cabling for a two instrument system with a scan platform, CLIST.DOC, is included with the document. This will need some modification to suit a specific flight.

If there is no platform then no Platform Motor Driver is needed.

A custom Power Filter Box is needed to match the power noise isolation requirements of the specific spacecraft.

2.0 Software Flight Tailoring

The software will be affected by the same considerations and will require adjustment in the following areas:

1. Platform Driver.
2. Data word length.
3. Coordination with multiple instruments.
4. Specific flight command response.

Again this does not represent a redo of the software but only tailoring for a specific flight.

The software included in the documentation has not been completely debugged on a flight microprocessor.

If the feature that words may be added in flight is to be implemented then modifications to the core are needed. These modifications have been developed but not debugged for other projects. This will probably be necessary if the RS-232 line is to be use in flight.

3.0 Hardware Completion

When work on the project was stopped due to loss of a specific flight opportunity, the following hardware components required completion or modification.

3.1 Microprocessor Board

The artwork for this PC board has been completed. We elected not to have PC boards made because minor changes to tailor to a new mission would then result in considerable added expense. If no such changes are made this artwork can be checked, run off on mylar, and sent out. A box for this board will have to be designed and fabricated.

3.2 CTS Board

One handbuild copy of the Calibration and Testing System board for the IBM PC is being held at RSI. An aditional copy will be needed if this system is to be used for testing the flight instrument.

3.3 Dual Stepper Motor Drive 480-224-0-2

Due to a design oversight this unit does not provide the needed direction control for one of the motors. This can be corrected by replacing U5, LS393, with two LS191 up/down counters. This can be done by adding a small dauther board to the existing units with the different chips. For high rel operation a new PC board will be needed.

3.4 Dual Motor Driver 490-224-0-2

Due to a design oversight this unit does not provide the needed direction control for the stepper motor. This can be corrected by replacing U5, LS393, with a LS191 up/down counter. This can be done by adding a small dauther board to the existing unit with the different chip. For high rel operation a new PC board will be needed. If the system has no platform, this unit will not be needed.

```
Screen # Ø
( FLIGHT PROGRAM   JOB 612                        JTR 13:12 Ø1/Ø5/89 )
( Last change:   Screen  ØØ1                      JTR 13:13 Ø1/Ø5/89 )

 ( APPENDIX XI -     EXAMPLE FLIGHT PROGRAM                          )
 (         Functionally identical with first instrument              )
 (                                                                   )
 (                               HUP2   Job 612                      )
 (                               8K PROM    UART                     )

 ( Version  1.Ø                                                      )
 ( J. T. Riley                                                       )
 ( 75.                                                               )

 ( File:  FP1.SCR       Disk:  HUP2   JOB 612  PROGRAM               )

Screen # 1
( 1 USER VARIABLE                                 JTR 13:13 Ø1/Ø5/89 )
FORTH DEFINITIONS    HEX

38 USER ELAP.T            ( ELAPSED TIME, DP, Data periods )
3A USER PRES.COM          ( Present mode command )
3C USER WAVE              ( Wavelength motor step count )
3E USER WAVE.C            ( Wavelength currently commanded )
4Ø USER ANG               ( Platform motor step count )
42 USER ANG.C             ( Platform step commanded )
44 USER H.V.C             ( High Voltage Status commanded )
46 USER DUST.C            ( Dust Cover position Commanded
48 USER ERR               ( Error flag )
4A USER SUN.P.OV.T        ( Sun Present Override Flat T-Over )
4C USER SP.TO             ( Sun Present Time Out, DP )
4E USER SHUT.C            ( Shutter Position Commanded, T=closed)


Screen # 2
( 2 CONSTANT                                      JTR 16:Ø2 Ø3/Ø1/88 )
HEX
5ØØØ CONSTANT DATA.BUF    ( Address of data buffer )
Ø0FE CONSTANT RSP         ( Return Stack Pointer Address )
DECIMAL
 1Ø   CONSTANT DATA.EN.R  ( Data Enable Rate, Hz )
  5   CONSTANT DATA#      ( Number of Data Bytes )
125   CONSTANT MOT.F      ( Stepper Motor Frequency, Hz )
 17   CONSTANT W.RELAY    ( Length relay pulse, msec )
4ØØ   CONSTANT TIME.OUT   ( Count limit for all motors, steps)
 1Ø   CONSTANT MOT.DEL    ( Motor Delay , seconds )
  6   CONSTANT WAVE/PLAT  ( Wavelength steps per Platform step)
1ØØ   CONSTANT PLAT2Ø     ( Platform steps in 2Ø Degrees )

 4Ø   CONSTANT BAUD       ( BAUD Rate constant, 4Ø - 48ØØ )
DATA.EN.R 6Ø * CONSTANT SP.D ( Sun Present Duration, Dat.En )


Laboratory Microsystems PC/FORTH 3.ØØ       12:07  10/21/89    fp1.scr
```

```
Screen # 3
( 3 CONSTANT                                          JTR 15:58 01/26/89 )
DECIMAL

100  CONSTANT WAVE1      ( Step for 1356 Angstroms )    ( ???)
120  CONSTANT WAVE2      ( Step for 1383 Angstroms )    ( ???)
140  CONSTANT WAVE3      ( Step for 1551 Angstroms )    ( ???)
```

```
Screen # 4
( 4 I.PORTS                                           JTR 15:09 01/06/89 )
ASSEMBLER  HEX

CODE  I.PORTS    (  --   . Initiate Ports )
        8 A MVI   44 OUT    ( Timer 2  /8 )
       C0 A MVI   45 OUT    ( Timer 2 cont Pulses )
       CD A MVI   40 OUT    ( A2-O, B2-I, C2-O )
       FF A MVI   64 OUT    ( Timer 3  /16384 )
       CF A MVI   65 OUT    ( Timer 3 cont Pulses )
       C[illegible] A MVI   60 OUT    ( A3-I, B3-I, C3-O )
     BAUD A MVI   84 OUT    ( Timer 4  /BAUD )
       40 A MVI   85 OUT    ( Timer 4 Sq wave )
       CE A MVI   80 OUT    ( A4-I, B4-O, C4-O )
       NEXT JMP C;
```

```
Screen # 5
( 5 I.VAR                                             JTR 13:59 04/12/88 )
HEX

: I.VAR  ( --   . Initialize Variables )
         0 0 ELAP.T D!      ( Zero Elapsed Time )
         SP.D @ 1+ 0 DMINUS  SP.TO D!  ( Sun Present Time Out )
         0 PRES.COM !    ( Present Command )
         0 ERR !         ( Error flag )
         0 H.V.C !       ( High Voltage Command )
         0 ERR !         ( Error Flag )
         0 SUN.P.OV.F !  ( Sun Present Override Flag )
         0 SHUT.C ! ;
```

```
Screen # 6
( 6 B2@, COM@. C4@                          JTR 15:39 02/19/88 )
HEX
CODE B2@ (  -- B, Read  Port B2 )
        42    IN        ( Read Port B2 )
        A  L  MOV       ( LSB )    0  H  MVI    ( MSB )
        HPUSH JMP  C: ( Place on stack )

CODE COM@ ( -- N . Read Command input, Port A4 )
        81    IN        ( Read Port A4)
        A  L  MOV       ( LSB )    0  H  MVI    ( MSB )
        HPUSH JMP  C: ( Place on stack )

CODE C4@ ( -- N . Read  Port C4 )
        83    IN        ( Read Port C4 )
        A  L  MOV       ( LSB )    0  H  MVI    ( MSB )
        HPUSH JMP  C: ( Place on stack )


Screen # 7
( 7 DATA.B.OUT . C2@ , C2!                  JTR 15:44 02/19/88 )
HEX
CODE DATA.B.OUT   ( B  -- . Data byte out, Port A2 )
        H    POP  L  A  MOV     ( Get From Stack )
        41 OUT   NEXT JMP  C: ( Out Port A2 )

CODE C2@ ( -- N . Read  Port C2 )
        41    IN       ( Read Port C2 )
        A  L  MOV       ( LSB )    0  H  MVI    ( MSB )
        HPUSH JMP  C: ( Place on stack )

CODE C2! ( N -- , Output Port C2 )
        H    POP  L  A  MOV     ( Get From Stack )
        41 OUT   NEXT JMP  C: ( Out Port C2 )



Screen # 8
( 8 C3! , MR! , MR@ . C4!                   JTR 15:41 02/19/88 )

CODE C3! ( N -- , Output Port C3 )
        H    POP  L  A  MOV     ( Get From Stack )
        63 OUT   NEXT JMP  C: ( Out Port C3 )
CODE MR! ( N -- . Output Motor Relay, Port B4 )
        H    POP  L  A  MOV     ( Get From Stack )
        82 OUT   NEXT JMP  C: ( Out Port B4 )
CODE MR@ ( -- N . Read  Port B4 )
        82    IN       ( Read Port B4 )
        A  L  MOV       ( LSB )    0  H  MVI    ( MSB )
        HPUSH JMP  C: ( Place on stack )

CODE C4! ( B -- . Output Port C4 )
        H    POP  L  A  MOV     ( Get From Stack )
        83 OUT   NEXT JMP  C: ( Out Port C4 )
```

```
Screen # 9
( 9 FID@ , ANA@ , T.L@ , O.DATA.C@          JTR 15:29 02/03/88 )
HEX

: FID@      ( -- N , Read Fiducials )
        C2@  2 OR  C2!      B2@ ;

: ANA@      (  -- N , Read Analog Levels )
        C2@  0FD AND  C2!   B2@ ;

: T.L@ ( -- N , Test Lamp Status, 0-Off, T-On )
        ANA@ 10 AND ;

: O.DATA.C@ ( -- N , Output Data Counter, 0-not/8, 20-/8 )
        ANA@ 20 AND ;



Screen # 10
( 10 STEP.MOT , MOT.WAIT , WAVE.HS , RESET JTR 09:48 02/05/88 )
HEX
: STEP.MOT  ( N --   , Step motor N once )
        C3!  1 MSEC 0 C3! ;

: MOT.WAIT  ( --   . Wait one motor cycle )
        1 MOT.F /  1000 * 1-   MSEC ;

: WAVE.HS     ( F --   . Wavelength motor, 0-Hold, 1-Step)
        1 AND  1+ MR!
        W.RELAY MSEC 0 MR! ;

: RESET.S.N.P  (  ---    ,   Reset Sun Not Present )
        C4@ DUP 8 AND C4! 10 MSEC  ( Pulse command )
        C4! ;


Screen # 11
( 11 PLAT.HS , DUST.HS , STEP.WAVE          JTR 09:51 02/05/88 )
HEX
: PLAT.HS     ( F --   , Scan Platform motor, 0-Hold, 1-Step)
        10 AND  10 + MR!
        W.RELAY MSEC  0 MR! ;

: DUST.HS     ( F --   . Dust Cover motor, 0-Hold. 1-Step)
        4 AND   4 + MR!
        W.RELAY MSEC 0 MR! ;

: STEP.WAVE    ( N --   . Step Wavelength N steps )
        ON WAVE.HS
        0 DO 4 STEP.MOT MOT.WAIT
        LOOP
        OFF WAVE.HS ;
```

```
Screen # 12
( 12 STEP.PLAT , STEP.DUST                          JTR 09:52 02/05/88 )

: STEP.PLAT    ( N --   , Scan Platform motor N steps )
        ON PLAT.HS
        0 DO  10 STEP.MOT  MOT.WAIT
        LOOP
        OFF PLAT.HS ;

: STEP.DUST    ( N --   , Step Dust Cover motor N steps )
        ON DUST.HS
        0 DO  8 STEP.MOT  MOT.WAIT
        LOOP
        OFF DUST.HS ;




Screen # 13
( 13 LATCH.POW , LATCH.DIR , T.LAMP , SHUT JTR 16:11 03/05/88 )

: LATCH.POW  (   F --   , Latch Motor Power   T-ON, F-OFF )
        40 AND    MR@ BF AND +    MR! :

: LATCH.DIR  ( F --- . Latch Motor Direction T-OPEN, F-CLOSE )
        80 AND    MR@ 7F AND +    MR! ;
: T.LAMP    ( F -- , Test Lamp Command. 0--Off, T-ON )
        1 AND C4@ FE AND +    C4! ;
: T.LAMP@   (    -- F , Test Lamp Indicator. 0-Off, T-On )
        ANA@ 10 AND FLAG ;
: SHUT   ( F -- , Shutter Command. 0-Open, T-Closed )
        2 AND C4@ FD AND +    C4! ;

: SHUT@ (    -- N , State of Shutter, 0-open. 2-closed)
        C4@ 2 AND ;


Screen # 14
( 14 S.C.I@ , S.C.OUT . D.O.B.L , COM.RS   JTR 15:44 01/06/89 )

: S.C.I@ ( -- F , Scan Complete ;Input, 0-No, T-Complete )
        ANA@ 80 AND ;

: S.C.OUT   ( F -- . Scan Complete Out, 0-Not. T-Complete )
        10 AND C4@ EF AND +    C4! ;

: D.O.B.L   (  -- , Data Out Buffer Load. Pulse )
        4  C4@  +  C4!       ( Set bit )
        C4@ FB AND C4! ;   ( Clear Bit )

: COM.RS    (  -- , Command Reset )
        20  C4@  +  C4!      ( Set bit )
        C4@ DF AND C4! ;   ( Clear Bit )
```

```
Screen # 15
( 15 WAVE.FID? , PLAT.FID? , DUST.O.FID?    JTR 15:59 01/06/89 )
HEX
: WAVE.FID?     (  -- F , At Wavelength Fid, T-at fid )
        FID@ 3 AND FLAG ;    ( Two Fids )
: PLAT.FID?     (  -- F , At Platform Fid, T-at fid )
        FID@ 30 AND FLAG  ;    ( Two Fids )

: DUST.O.FID?     (  -- F , At Dust Cover Open Fid, T-Open)
        FID@  4 AND FLAG  ;
: DUST.C.FID?     (  -- F , At Dust Cover Closed Fid, T-Closed)
        FID@  8 AND FLAG  ;

: LATCH.O.FID?     (  -- F , At Latch Open Fid, T-Open)
        FID@ 40 AND FLAG  ;
: LATCH.C.FID?     (  -- F , At Latch Closed Fid, T-Closed)
        FID@ 40 AND FLAG  ;


Screen # 16
( 16 N.COM? , SEC                                JTR 16:13 03/05/88 )

: N.COM?    (  -- F,  New Command ? , T-NEW )
        COM@
        PRES.COM @ =  NOT ;   ( Does not change command )

: SEC        (  N  --  , Wait Seconds )
        0 DO ELAP.T D@ DATA.EN.R @ 0 D+
                BEGIN
                   2DUP ELAP.T D@ D= N.COM? OR UNTIL
        DROP DROP LOOP ;

: HV@   (  ---- F , High Voltage Status 0-low, T- Good )
        ANA@ 2 AND FLAG ;


Screen # 17
( 17 STEP.WAVE.FID , STEP.PLAT.FID             JTR 09:53 02/05/88 )

: STEP.WAVE.FID    ( --   , Step to Wavelength Fiducial )
        TIME.OUT
        BEGIN 1- DUP  WAVE.FID? NOT AND  WHILE  ( At fid ? )
           1 STEP.WAVE REPEAT DROP ;     ( Step to fid )

: STEP.PLAT.FID    ( --   , Step to Scan Platform Fiducial )
        TIME.OUT
        BEGIN 1- DUP  PLAT.FID? NOT AND WHILE  ( At fid ? )
           1 STEP.PLAT REPEAT DROP ;     ( Step to fid )
```

```
Screen # 18
( 18 S.DUST.O.FID . S.DUST.C.FID , DUST     JTR 10:24 02/05/88 )

: S.DUST.O.FID (   --   , Open Dust Cover with time out )
        TIME.OUT
        BEGIN 1- DUP  DUST.O.FID? NOT AND WHILE
           1 STEP.DUST REPEAT DROP ;     ( Step to fid )

: S.DUST.C.FID (   --   , CLOSE Dust Cover with time out )
        TIME.OUT
        BEGIN 1- DUP  DUST.C.FID? NOT AND WHILE
           1 STEP.DUST REPEAT DROP ;     ( Step to fid )

: DUST    ( F --  , Open or Close Dust Cover, T-Open )
        IF     S.DUST.O.FID
        ELSE   S.DUST.C.FID THEN ;



Screen # 19
( 19 WAIT.DATA . CLOSE.LATCH , OPEN.LATCH  JTR 09:54 02/05/88 )

: WAIT.DATA (    --  Wait for one data out )
        ELAP.T @ 1+
        BEGIN DUP ELAP.T @ = UNTIL DROP ;

: CLOSE.LATCH   (   --   . Close latch )
        STEP.PLAT.FID
        OFF LATCH.DIR    ON LATCH.POW
        BEGIN LATCH.C.FID? UNTIL
        OFF LATCH.POW ;
: OPEN.LATCH   (   --   , Open latch )
        ON LATCH.DIR    ON LATCH.POW
        BEGIN LATCH.O.FID? UNTIL
        OFF LATCH.POW ;



Screen # 20
( 20 SCAN.PLAT.WAVE                             JTR 13:59 24/12/88 )

: SCAN.PLAT.WAVE   ( N --  , Scan Platform and Wavelength )
        0 DO                      ( Multiple Scans )
           TIME.OUT
           BEGIN 1- DUP  PLAT.FID? NOT AND WHILE
              1 STEP.PLAT          ( Step Platform )
              WAVE/PLAT 0 DO       ( Step Wavelength )
                1 STEP.WAVE   WAIT.DATA  ( Wait for Data Out )
              LOOP                ( Next Wavelength step )
           REPEAT DROP            ( Next Platform step )
        LOOP ;                    ( Next Scan )
```

```
Screen # 21
( 21 SCAN.WAVE                                    JTR 10:11 02/05/88 )

: SCAN.WAVE  ( N --   , Scan in Wavelength only )
       0 DO                          ( Multiple Scans )
          TIME.OUT
          BEGIN 1- DUP  WAVE.FID? NOT AND WHILE
               1 STEP.WAVE   WAIT.DATA  ( Wait for Data Out )
          REPEAT DROP                ( Next Wavelength step )
       LOOP ;                        ( Next Scan )



Screen # 22
( 22 SCAN20 . SCAN.PLAT . SAVE.SYS              JTR 10:11 02/05/88 )

: SCAN20  (  --   . 1/2 Scan of Platform only  )
          PLAT20 0 DO
             1 STEP.PLAT             ( Step Platform )
               WAIT.DATA             ( Wait for Data Out )
          LOOP ;                     ( Next Platform Step )

: SCAN.PLAT  ( N --   . Scan Platform only )
       0 DO
          SCAN20 SCAN20 STEP.PLAT.FID
       LOOP ;

: SAVE.SYS  (    --    . Save System state )
       WAVE @ WAVE.C !   ANG @ ANG.C ! ;


Screen # 23
( 23 COM.CK                                       JTR 16:28 02/01/88 )

: COM.CK  (  -- N , Check for New Command )
       N.COM?  DUP  IF
          10 SEC
             N.COM? IF
          DROP COM@ DUP PRES.COM ' D.O.B.L
       THEN THEN ;
```

```
Screen # 24
( 24 SCAN1                                          JTR 14:00 04/12/88 )

: SCAN1  (    ---   , Scan Mode 1 command )
        STEP.PLAT.FID       ( Step Platform the fiducial )
        STEP.WAVE.FID       ( Step Wavelength to fiducial )
        BEGIN               ( Start repetitive scans )
           WAVE1 STEP.WAVE   ( Step to Wavelength 1 )
           SCAN20            ( Scan Platform 1/2 cycle )
           WAVE2 STEP.WAVE   ( Step to Wavelength 2 )
           SCAN20            ( Scan Platform 1/2 cycle )
           STEP.WAVE.FID     ( Step Wavelength to fiducial)
           STEP.PLAT.FID     ( Step Platform to fiducial)
           1 SCAN.PLAT.WAVE ( Scan Platform and wavelength)
        N.COM? UNTIL ;   ( Check for New command )


Screen # 25
( 25 SCAN2                                          JTR 10:11 02/05/88 )

: SCAN2 (   --   . Scan Mode 2 command )
        STEP.PLAT.FID         ( Step Platform to Fiducial )
        STEP.WAVE.FID         ( Step Wavelength to Fiducial )
        BEGIN
          1 SCAN.PLAT.WAVE     ( Scan Platform and Wavelength )
          N.COM? UNTIL         ( Check for new Command )
        ;


Screen # 26
( 26 SCAN3                                          JTR 14:00 04/12/88 )

: SCAN3 (   --   . Scan Mode 3 command )
        STEP.PLAT.FID        ( Set Platform to Fiducial )
        STEP.WAVE.FID        ( Step Wavelength to Fiducial )
        WAVE3  STEP.WAVE     ( Step Wavelength to Wave 3 )
        BEGIN                ( Wait for new command )
          N.COM? UNTIL
        ;
```

```
Screen # 27
( 27 SUN? , SUN.CK                              JTR 16:23 03/05/88 )
: SUN.N.P@ (  -- F . Sun Not Present Bit 0-Alarm, 1- Safe )
        ANA@ 40 AND FLAG ;
: SUN?    (  -- F , Sun Present , 0-Safe, T-Alarm )
        SUN.P@  NOT DUP
           IF ELAP.T D@ SP.TO D! THEN    ( Set Time Out )
        SP.TO  D@  SP.D  0 D+  ELAP.T D@  D>
        OR ;

: SUN.CK    (  --   , Close Shutter if sun present )
        SUN?  IF
           NOOP       ( Action is automatic )
          ELSE
           RESET.S.N.P  ( Reset Sun Not Present )
        THEN ;
```

```
Screen # 28
( 28 H.V . NSP.OV . WT.OPT.T.I                  JTR 10:37 02/05/88 )
HEX
: H.V    (  F ---   , PTM High Voltage T-ON / F-auto )
        IF  C4@ 4 AND C4!    ( Sun Present Overide Bit )
        ELSE  C4@ FB AND C4! THEN ;

: NSP.OV  (  ---   . Override Sun Present Command )
        ON SUN.P.OV.F  !    ( Set Sun Present Override Flag )
        ON H.V         ;    ( Set Sun Present Override Bit )

: WT.OPT.T.I  (   --    , Wait for Optical Test Indicator )
         BEGIN
            ANA@ 10 AND N.COM? OR  NOT WHILE
         SUN.CK REPEAT ;
```

```
Screen # 29
( 29 FCAL                                       JTR 10:37 02/05/88 )

: FCAL (  --  , Flight Calibration Command )
        ON T.LAMP          ( Test lamp on )
        SAVE.SYS           ( Remember system state )
        CLOSE  DUST        ( Close Dust Cover )
        ON    H.V          ( Force HV on )
        STEP.WAVE.FID      ( Step to Wavelength Fid )
        5 SCAN.WAVE        ( 5 Wavelength Scans )
        WT.OPT.T.I         ( Wait for Optical Test Indicator )
        5 SCAN.WAVE        ( 5 Wavelength Scans )
        OFF T.LAMP         ( Test Lamp Off )
        DUST.C @ DUST      ( Return Dust Cover )
        WAVE.C @ STEP.WAVE ; ( Return Wavelength )
```

```
Screen # 30
( 30 STAND.ST , OP.LAT                              JTR 10:37 02/05/88 )

: STAND.ST  (  --  , Standard Start Configuration command )
        STEP.WAVE.FID     ( Step Wavelength to fiducial )
        S.DUST.O.FID      ( Open Dust cover )
        OPEN SHUT         ( Open Shutter )
        OPEN SHUT.C !
        OFF T.LAMP        ( Test Lamp Off )
        ON H.V            ( High voltage ON )
        ;

: OP.LAT  (  --  , Open Latch command )
        OPEN.LATCH           ( Open Latch )
        STEP.PLAT.FID :      ( Step Platform to Fiducial )



Screen # 31
( 31 SHT.DW                                         JTR 10:37 02/05/88 )

: SHT.DW  (  --  , Normal Shut Down command )
        STEP.WAVE.FID        ( Step wavelength to fiducial )
        STEP.PLAT.FID        ( Step Platform to fiducial )
        OPEN SHUT            ( Open Shutter )
        OPEN SHUT.C !
        OFF T.LAMP           ( Test Lamp off )
        CLOSE.LATCH          ( Close latch )
        ;


        ( Turn off high voltage ???)



Screen # 32
( 32 SER.COM.                                       JTR 16:17 03/01/88 )
HEX
: SER.COM   (  --  , Service incoming commands )
    BEGIN  1 WHILE            ( Infinite loop )
        COM.CK                ( Get new command )
            DUP 1 = IF SCAN1 THEN
            DUP 2 = IF SCAN2 THEN
            DUP 4 = IF SCAN3 THEN
            DUP 8 = IF NSP.DV THEN
            DUP 10 = IF FCAL THEN
            DUP 20 = IF STAND.ST THEN
            DUP 40 = IF OP.LAT THEN
                80 = IF SHT.DW THEN
        SUN?  STAND.ST
    REPEAT :     ( Infinite loop )
```

```
Screen # 33
(                                         JTR 10:36 02/05/88 )
```

```
Screen # 34
( 34 CT@                                  JTR 15:45 02/19/88 )

CODE CT@ ( -- N . Read Detector count. Port A3, B3  )
        0  A   MVI
        63     OUT       ( Stop Count )
        61     IN        ( Read Port A6, LSB )
        A  L   MOV       ( LSB )
        62     IN        ( Read Port B6 )
        A  H   MOV       ( MSB )
        2  A   MVI
        63     OUT       ( Reset Counter )
        1  A   MVI
        63     OUT       ( Enable Counter )
        HPUSH JMP  C;  ( Place on stack )
```

```
Screen # 35
( 35 DATA.IN                              JTR 16:31 03/05/88 )
: DATA.IN  ( --  , Read new Data, Store at DATA.BUF )
     CT@ DUP E000 AND  IF 1 ERR ! THEN 1FFF AND  ( Count )
        PRES.COM 2000 * + DATA.BUF !        ( Command )
        WAVE 03FF AND                       ( Wavelength )
        WAVE.FID? 400 AND +                 ( Wave Fid )
        SHUT@     800 AND +                 ( Shutter )
        HV@      1000 AND +                 ( High Voltage )
        T.LAMP@  2000 AND +                 ( Test Lamp )
        HV@      4000 AND +                 ( High Voltage )
        SUN.N.P@ 8000 AND + 2 DATA.BUF + !  ( Sun Not Present )
        ANG        1FFF AND                 ( Platform Angle )
        PLAT.FID?  2000 AND +               ( Platform Fid )
        LATCH.FID? 4000 AND +               ( Latch Status )
        ERR 8000  AND  + 4 DATA.BUF + ! ;    ( Error )
```

```
Screen # 36
( 36 DATA.OUT , Data handling in Forth       JTR 14:01 04/12/88 )

: DATA.OUT   (   --   , Data service on interrupt 7.5   )
        DATA# 0 DO
           BEGIN  O.DATA.C@ UNTIL        ( Wait for counter )
           DATA.BUF I  + C!  DATA.B.OUT  ( Output data byte )
         LOOP
        DATA.IN                  ( Read new Data   )
        DATA.BUF @ DATA.B.OUT     ( Preload output )
        ELAP.T D@ 1 0 D+ ELAP.T D!   ( Increment elapsed time )
        IRET :                 ( End interrupt )
```

```
Screen # 37
( 37 INT7.5                                    JTR 15:12 02/19/88 )
HEX
CODE INT7.5  ( -- ,  Set up Forth interrupt 7.5 )
        PSW PUSH B PUSH D PUSH H PUSH
        RSP  LHLD  H PUSH  ( Save Return Stack Pointer )
        ' DATA.OUT CFA H LXI     ( Address of word DATA.OUT )
        0072 JMP         ( NEXT1 . Mimic EXECUTE )
        C:
```

```
Screen # 38
( 38 Initiate Interrupt 7.5. INIT                  12:53 01/05/89 )
HEX
        C3       2002 C!                ( JMP command )
        '        INT7.5  2003 !  ( Upper interrupt Vector )

: INIT   (  --  , Initiate System )
        4000 DP !                 ( Set in flight HERE )
        DIS.I  I.VAR  I.PORTS       ( Initiate system )
        SUN.CK                    ( Check Sun Present )
        10000 MSEC  ." FP " CR    ( Wait 10 seconds, sign on )
        EI7.5&5.5 ( EI7.5 )       ( Enable Data )
        SER.COM :                 ( Service incoming Commands )

    (   ' INIT  CFA  2000 ! ) ( Cold Start vector )
          ( **  Set Cold Start Vector before burning PROM ** )
  ( End of Flight Program )

Laboratory Microsystems PC/FORTH 3.00      12:12  10/21/89   fp1.scr
```

```
Screen # 39
( Testing only                                JTR 11:57 12/16/87 )
 HEX

 4000 HERE - DUP 64 2000 */
 DECIMAL CR SPACE . ." % " . ." Bytes remaining "  CR
```

```
Screen # 40
( LEAVE BLANK                                 JTR 15:58 08/10/87 )
```

```
Screen # 41
```

```
 0 ( FLIGHT PROGRAM  JOB 612                       JTR 13:12 01/05/89 )
 1 ( 1 USER VARIABLE                               JTR 13:13 01/05/89 )
 2 ( 2 CONSTANT                                    JTR 16:02 03/01/88 )
 3 ( 3 CONSTANT                                    JTR 15:58 01/06/89 )
 4 ( 4 1.PORTS                                     JTR 15:09 01/06/89 )
 5 ( 5 I.VAR                                       JTR 13:59 04/12/88 )
 6 ( 6 B2@, COM@ C4@                               JTR 15:39 02/19/88 )
 7 ( 7 DATA.B.OUT , C2@ , C2!                      JTR 15:44 02/19/88 )
 8 ( 8 C3! , MR! , MR@ , C4!                       JTR 15:41 02/19/88 )
 9 ( 9 FID@ , ANA@ , T.L@ , O.DATA.C@              JTR 15:29 02/03/88 )
10 ( 10 STEP.MOT , MOT.WAIT , WAVE.HS , RESET JTR 09:48 02/05/88 )
11 ( 11 PLAT.HS , DUST.HS , STEP.WAVE              JTR 09:51 02/05/88 )
12 ( 12 STEP.PLAT , STEP.DUST                      JTR 09:52 02/05/88 )
13 ( 13 LATCH.POW , LATCH.DIR , T.LAMP , SHUT JTR 16:11 03/05/88 )
14 ( 14 S.C.I@ , S.C.OUT , D.O.B.L , COM.RS        JTR 15:44 01/06/89 )
15 ( 15 WAVE.FID? , PLAT.FID? , DUST.O.FID?        JTR 15:59 01/06/89 )
16 ( 16 N.COM? , SEC                               JTR 16:13 03/05/88 )
17 ( 17 STEP.WAVE.FID , STEP.PLAT.FID              JTR 09:53 02/05/88 )
18 ( 18 S.DUST.O.FID , S.DUST.C.FID , DUST         JTR 10:24 02/05/88 )
19 ( 19 WAIT.DATA , CLOSE.LATCH , OPEN.LATCH       JTR 09:54 02/05/88 )
20 ( 20 SCAN.PLAT.WAVE                             JTR 13:59 04/12/88 )
21 ( 21 SCAN.WAVE                                  JTR 10:11 02/05/88 )
22 ( 22 SCAN20 , SCAN.PLAT , SAVE.SYS              JTR 10:11 02/05/88 )
ok


22 ( 22 SCAN20 , SCAN.PLAT , SAVE.SYS              JTR 10:11 02/05/88 )
ok
23 44 INDEX

23 (   23 COM.CK                                   JTR 16:28 03/01/88 )
24 (   24 SCAN1                                    JTR 14:00 04/12/88 )
25 (   25 SCAN2                                    JTR 10:11 02/05/88 )
26 (   26 SCAN3                                    JTR 14:00 04/12/88 )
27 (   27 SUN? , SUN.CK                            JTR 16:23 03/05/88 )
28 (   28 H.V , NSP.OV , WT.OPT.T.I                JTR 10:37 02/05/88 )
29 (   29 FCAL                                     JTR 10:37 02/05/88 )
30 (   30 STAND.ST , OP.LAT                        JTR 10:37 02/05/88 )
31 (   31 SHT.DW                                   JTR 10:37 02/05/88 )
32 (   32 SER.COM,                                 JTR 16:17 03/01/88 )
33 (                                               JTR 10:36 02/05/88 )
34 (   34 CT@                                      JTR 15:45 02/19/88 )
35 (   35 DATA.IN                                  JTR 16:31 03/05/88 )
36 (   36 DATA.OUT , Data handling in Forth        JTR 14:01 04/12/88 )
37 (   37 INT7.5                                   JTR 15:12 02/19/88 )
38 (   38 Initiate Interrupt 7.5, INIT                 12:53 01/05/89 )
39 (   Testing only                                JTR 11:57 12/16/87 )
40 (   LEAVE BLANK                                 JTR 15:58 08/10/87 )
41
ok
```

```
Screen # 0
( FLIGHT PROGRAM  MASTER    JOB 612           JTR 12:54 01/05/89 )
( Last change:   Screen  038                  JTR 12:58 01/05/89 )

 ( APPENDIX XII -  MASTER UNIT FLIGHT PROGRAM                      )
 (                   Master unit with platform                     )
 (                                                                 )
 (                                SACS    Job 612                  )
 (                                8K PROM    UART                  )

 ( Version  1.0                                                    )
 ( J. T. Riley                                                     )
 ( RSI                                                             )

 ( File:  FPM.SCR        Disk:  HUP2   JOB 612   PROGRAM           )

Screen # 1
( 1 USER VARIABLE                             JTR 12:57 01/05/89 )
FORTH DEFINITIONS   HEX

36 USER ELAP.T            ( ELAPSED TIME, DP, Data periods )
3A USER PRES.COM          ( Present mode command )
3C USER WAVE              ( Wavelength motor step count )
3E USER WAVE.C            ( Wavelength currently commanded )
40 USER ANG               ( Platform motor step count )
42 USER ANG.C             ( Platform step commanded )
44 USER H.V.C             ( High Voltage Status commanded )
46 USER DUST.C            ( Dust Cover position Commanded )
48 USER ERR               ( Error flag )
4A USER SUN.P.OV.F        ( Sun Present Override Flat T-Over )
4C USER SP.TO             ( Sun Present Time Out, DP )
4E USER SHUT.C            ( Shutter Position Commanded, T-closed)


Screen # 2
( 2 CONSTANT                                  JTR 16:02 03/01/88 )
HEX
6000 CONSTANT DATA.BUF    ( Address of data buffer )
80FE CONSTANT RSP         ( Return Stack Pointer Address )
DECIMAL
 10   CONSTANT DATA.EN.R  ( Data Enable Rate, Hz )
  5   CONSTANT DATA#      ( Number of Data Bytes )
125   CONSTANT MOT.F      ( Stepper Motor Frequency, Hz )
 10   CONSTANT W.RELAY    ( Length relay pulse, msec )
400   CONSTANT TIME.OUT   ( Count limit for all motors, steps)
 10   CONSTANT MOT.DEL    ( Motor Delay , seconds )
  6   CONSTANT WAVE/PLAT  ( Wavelength steps per Platform step)
100   CONSTANT PLAT20     ( Platform steps in 20 Degrees )

 40   CONSTANT BAUD       ( BAUD Rate constant, 40 - 4800 )
DATA.EN.R 60 * CONSTANT SP.D ( Sun Present Duration, Dat.En )

Laboratory Microsystems PC/FORTH 3.00      12:40   10/21/89    fpm.scr
```

```
Screen # 3
( 3 CONSTANT                                        JTR 15:50 01/06/89 )
DECIMAL

100  CONSTANT WAVE1      ( Step for 1356 Angstroms )    ( ???)
120  CONSTANT WAVE2      ( Step for 1383 Angstroms )    ( ???)
140  CONSTANT WAVE3      ( Step for 1551 Angstroms )    ( ???)
```

```
Screen # 4
( 4 I.PORTS                                         JTR 15:09 01/06/89 )
ASSEMBLER  HEX

CODE  I.PORTS    (  --  . Initiate Ports )
        8 A MVI  44 OUT    ( Timer 2  /8 )
       C7 A MVI  45 OUT    ( Timer 2 cont Pulses )
       CD A MVI  40 OUT    ( A2-O, B2-I, C2-O )
       FF A MVI  64 OUT    ( Timer 3  /16384 )
       CF A MVI  65 OUT    ( Timer 3 cont Pulses )
       CD A MVI  60 OUT    ( A3-I, B3-I, C3-O )
     BAUD A MVI  84 OUT    ( Timer 4  /BAUD )
       40 A MVI  85 OUT    ( Timer 4 Sq wave )
       CE A MVI  80 OUT    ( A4-I, B4-O, C4-O )
       NEXT JMP C;
```

```
Screen # 5
( 5 I.VAR                                           JTR 14:04 04/12/88 )
HEX

: I.VAR   ( --  . Initialize Variables )
        0 0 ELAP.T D!       ( Zero Elapsed Time )
        SP.D @ 1+ 0 DMINUS  SP.TO D!  ( Sun Present Time Out )
        0 PRES.COM !    ( Present Command )
        0 ERR !         ( Error flag )
        0 H.V.C !       ( High Voltage Command )
        0 ERR !         ( Error Flag )
        0 SUN.P.OV.F !  ( Sun Present Override Flag )
        0 SHUT.C ! ;
```

```
Screen # 6
( 6 B2@, COM@, C4@                              JTR 15:39 02/19/88 )
HEX
CODE B2@ (  -- B, Read  Port B2 )
         42     IN        ( Read Port B2 )
         A  L  MOV        ( LSB )    0  H  MVI    ( MSB )
         HPUSH JMP  C: ( Place on stack )

CODE COM@ ( -- N , Read Command input, Port A4 )
         81     IN        ( Read Port A4)
         A  L  MOV        ( LSB )    0  H  MVI    ( MSB )
         HPUSH JMP  C: ( Place on stack )

CODE C4@ ( -- N , Read  Port C4 )
         83     IN        ( Read Port C4 )
         A  L  MOV        ( LSB )    0  H  MVI    ( MSB )
         HPUSH JMP  C: ( Place on stack )


Screen # 7
( 7 DATA.B.OUT , C2@ , C2!                      JTR 15:44 02/19/88 )
HEX
CODE DATA.B.OUT    ( B  -- , Data byte out, Port A2 )
         H    POP  L  A  MOV     ( Get From Stack )
         41 OUT    NEXT JMP  C: ( Out Port A2 )

CODE C2@ ( -- N . Read  Port C2 )
         41     IN        ( Read Port C2 )
         A  L  MOV        ( LSB )    0  H  MVI    ( MSB )
         HPUSH JMP  C: ( Place on stack )

CODE C2! ( N -- , Output Port C2 )
         H    POP  L  A  MOV     ( Get From Stack )
         41 OUT    NEXT JMP  C: ( Out Port C2 )



Screen # 8
( 8 C3! , MR! , MR@ , C4!                       JTR 15:41 02/19/88 )

CODE C3! ( N -- , Output Port C3 )
         H    POP  L  A  MOV     ( Get From Stack )
         63 OUT    NEXT JMP  C: ( Out Port C3 )
CODE MR! ( N -- , Output Motor Relay, Port B4 )
         H    POP  L  A  MOV     ( Get From Stack )
         82 OUT    NEXT JMP  C: ( Out Port B4 )
CODE MR@ ( -- N , Read  Port B4 )
         82     IN        ( Read Port B4 )
         A  L  MOV        ( LSB )    0  H  MVI    ( MSB )
         HPUSH JMP  C: ( Place on stack )

CODE C4! ( B -- , Output Port C4 )
         H    POP  L  A  MOV     ( Get From Stack )
         83 OUT    NEXT JMP  C: ( Out Port C4 )
```

```
Screen # 9
( 9 FID@ , ANA@ , T.L@ , O.DATA.C@          JTR 15:29 02/03/88 )
HEX

: FID@      ( -- N , Read Fiducials )
        C2@  2 OR  C2!      B2@ ;

: ANA@      (  -- N , Read Analog Levels )
        C2@  0FD AND  C2!    B2@ ;

: T.L@ ( -- N , Test Lamp Status, 0-Off, T-On )
        ANA@ 10 AND ;

: O.DATA.C@ ( -- N , Output Data Counter, 0-not/8. 20-/8 )
        ANA@ 20 AND ;



Screen # 10
( 10 STEP.MOT , MOT.WAIT , WAVE.HS . RESET JTR 09:48 02/05/88 )
HEX
: STEP.MOT  ( N --   , Step motor N once )
        C3!  1 MSEC 0 C3! ;

: MOT.WAIT  ( --   , Wait one motor cycle )
        1 MOT.F /  1000 * 1-   MSEC ;

: WAVE.HS    ( F --   , Wavelength motor, 0-Hold, 1-Step)
        1 AND  1+ MR!
        W.RELAY MSEC 0 MR! ;

: RESET.S.N.P  (  ---    .  Reset Sun Not Present )
        C4@ DUP 8 AND C4! 10 MSEC  ( Pulse command )
        C4! ;



Screen # 11
( 11 PLAT.HS , DUST.HS , STEP.WAVE          JTR 09:51 02/05/88 )
HEX
: PLAT.HS    ( F --   , Scan Platform motor, 0-Hold. 1-Step)
        10 AND  10 + MR!
        W.RELAY MSEC  0 MR! ;

: DUST.HS    ( F --   . Dust Cover motor, 0-Hold, 1-Step)
        4 AND  4 + MR!
        W.RELAY MSEC 0 MR! ;

: STEP.WAVE   ( N --   , Step Wavelength N steps )
        ON WAVE.HS
        0 DO 4 STEP.MOT MOT.WAIT
        LOOP
        OFF WAVE.HS ;
```

```
Screen # 12
( 12 STEP.PLAT . STEP.DUST                        JTR 09:52 02/05/88 )

: STEP.PLAT    ( N --   , Scan Platform motor N steps )
        ON PLAT.HS
        0 DO  10 STEP.MOT  MOT.WAIT
        LOOP
        OFF PLAT.HS ;

: STEP.DUST    ( N --   , Step Dust Cover motor N steps )
        ON DUST.HS
        0 DO  8 STEP.MOT  MOT.WAIT
        LOOP
        OFF DUST.HS ;



Screen # 13
( 13 LATCH.POW . LATCH.DIR . T.LAMP . SHUT JTR 16:11 03/03/88 )

: LATCH.POW  (  F --   . Latch Motor Power   T-ON, F-OFF )
        40 AND   MR@ BF AND +    MR! ;

: LATCH.DIR  ( F ---  , Latch Motor Direction T-OPEN, F-CLOSE )
        80 AND   MR@ 7F AND +    MR! ;
: T.LAMP    ( F --  , Test Lamp Command, 0-Off, T-ON )
        1 AND C4@ FE AND +    C4! ;
: T.LAMP@  (    -- F , Test Lamp Indicator, 0-Off, T-On )
        ANA@ 10 AND FLAG ;
: SHUT  ( F -- , Shutter Command. 0-Open, T-Closed )
        2 AND C4@ FD AND +    C4! ;

: SHUT@ (    -- N . State of Shutter, 0-open, 2-closed)
        C4@ 2 AND ;


Screen # 14
( 14 S.C.I@ . S.C.OUT . D.O.B.L . COM.RS   JTR 15:44 01/06/89 )

: S.C.I@ ( -- F . Scan Complete ;Input, 0-No, T-Complete )
        ANA@ 80 AND ;

: S.C.OUT  ( F -- . Scan Complete Out, 0-Not, T-Complete )
        10 AND C4@ EF AND +    C4! ;

: D.O.B.L   (   -- , Data Out Buffer Load, Pulse )
        4  C4@  +  C4!     ( Set bit )
        C4@ FB AND C4! ;  ( Clear Bit )

: COM.RS    (   -- , Command Reset )
        20  C4@  +  C4'     ( Set bit )
        C4@ DF AND C4! ;  ( Clear Bit )


Laboratory Microsystems PC/FORTH 3.00       12:51   10/21/89    fom.scr
```

```
Screen # 15
( 15 WAVE.FID? , PLAT.FID? , DUST.O.FID?    JTR 15:59 01/06/89 )
HEX
: WAVE.FID?     (  -- F , At Wavelength Fid, T-at fid )
        FID@ 3 AND FLAG ;    ( Two Fids )
: PLAT.FID?     (  -- F , At Platform Fid, T-at fid )
        FID@ 30 AND FLAG  :    ( Two Fids )

: DUST.O.FID?     (  -- F , At Dust Cover Open Fid, T-Open)
        FID@  4 AND FLAG  ;
: DUST.C.FID?     (  -- F , At Dust Cover Closed Fid, T-Closed)
        FID@  8 AND FLAG  ;

: LATCH.O.FID?      (  -- F , At Latch Open Fid, T-Open)
        FID@ 40 AND FLAG  ;
: LATCH.C.FID?      (  -- F , At Latch Closed Fid, T-Closed)
        FID@ 40 AND FLAG  ;


Screen # 16
( 16 N.COM? . SEC                                 JTR 16:13 03/05/88 )

: N.COM?    (  -- F,  New Command ? . T-NEW )
        COM@
        PRES.COM @ =  NOT ;  ( Does not change command )

: SEC         (  N  --  . Wait Seconds )
        0 DO ELAP.T D@ DATA.EN.R @ 0 D+
                BEGIN
                   2DUP ELAP.T D@ D= N.COM? OR UNTIL
        DROP DROP LOOP ;

: HV@   (  ---  F , High Voltage Status 0-low, T- Good )
        ANA@ 2 AND FLAG ;



Screen # 17
( 17 STEP.WAVE.FID . STEP.PLAT.FID              JTR 09:53 02/05/88 )

: STEP.WAVE.FID    ( --   . Step to Wavelength Fiducial )
        TIME.OUT
        BEGIN 1- DUP  WAVE.FID? NOT AND  WHILE  ( At fid ? )
           1 STEP.WAVE REPEAT DROP ;     ( Step to fid )

: STEP.PLAT.FID    ( --   . Step to Scan Platform Fiducial )
        TIME.OUT
        BEGIN 1- DUP  PLAT.FID? NOT AND WHILE  ( At fid ? )
           1 STEP.PLAT REPEAT DROP :     ( Step to fid )
```

```
Screen # 18
( 18 S.DUST.O.FID , S.DUST.C.FID , DUST     JTR 10:24 02/05/88 )

: S.DUST.O.FID (  --   . Open Dust Cover with time out )
        TIME.OUT
        BEGIN 1- DUP  DUST.O.FID? NOT AND WHILE
           1 STEP.DUST REPEAT DROP ;     ( Step to fid )

: S.DUST.C.FID (  --   . CLOSE Dust Cover with time out )
        TIME.OUT
        BEGIN 1- DUP  DUST.C.FID? NOT AND WHILE
           1 STEP.DUST REPEAT DROP ;     ( Step to fid )

: DUST   ( F --   . Open or Close Dust Cover. T=Open )
        IF    S.DUST.O.FID
        ELSE  S.DUST.C.FID THEN ;


Screen # 19
( 19 WAIT.DATA . CLOSE.LATCH , OPEN.LATCH  JTR 09:54 02/05/88 )

: WAIT.DATA (    --  Wait for one data out )
        ELAP.T @ 1+
        BEGIN DUP ELAP.T @ = UNTIL DROP ;

: CLOSE.LATCH  (  --   . Close latch )
        STEP.PLAT.FID
        OFF LATCH.DIR   ON LATCH.POW
        BEGIN LATCH.C.FID? UNTIL
        OFF LATCH.POW ;
: OPEN.LATCH  (  --   . Open latch )
        ON LATCH.DIR   ON LATCH.POW
        BEGIN LATCH.O.FID? UNTIL
        OFF LATCH.POW ;


Screen # 20
( 20 SCAN.PLAT.WAVE                         JTR 14:05 04/12/88 )

: SCAN.PLAT.WAVE  ( N --   . Scan Platform and Wavelength )
        0 DO                     ( Multiple Scans )
           TIME.OUT
           BEGIN 1- DUP  PLAT.FID? NOT AND WHILE
              1 STEP.PLAT          ( Step Platform )
              WAVE/PLAT 0 DO       ( Step Wavelength )
                1 STEP.WAVE   WAIT.DATA  ( Wait for Data Out )
              LOOP                 ( Next Wavelength step )
           REPEAT DROP             ( Next Platform step )
        LOOP ;                     ( Next Scan )
```

```
Screen # 21
( 21 SCAN.WAVE                                     JTR 10:11 02/05/88 )

: SCAN.WAVE   ( N --   . Scan in Wavelength only )
        0 DO                          ( Multiple Scans )
          TIME.OUT
          BEGIN 1- DUP  WAVE.FID? NOT AND WHILE
                1 STEP.WAVE    WAIT.DATA  ( Wait for Data Out )
          REPEAT DROP                 ( Next Wavelength step )
        LOOP ;                        ( Next Scan )
```

```
Screen # 22
( 22 SCAN20 , SCAN.PLAT , SAVE.SYS             JTR 10:11 02/05/88 )

: SCAN20   (  --   . 1/2 Scan of Platform only   )
           PLAT20 0 DO
              1 STEP.PLAT            ( Step Platform )
                WAIT.DATA            ( Wait for Data Out )
           LOOP ;                    ( Next Platform Step )

: SCAN.PLAT   ( N --   . Scan Platform only )
        0 DO
          SCAN20 SCAN20 STEP.PLAT.FID
        LOOP ;

: SAVE.SYS   (    --     . Save System state )
        WAVE @ WAVE.C !    ANG @ ANG.C ! ;
```

```
Screen # 23
( 23 COM.CK                                      JTR 16:28 03/01/88 )

: COM.CK   (  -- N . Check for New Command )
        N.COM?  DUP  IF
          10 SEC
             N.COM? IF
          DROP COM@ DUP PRES.COM ! D.O.B.L
        THEN THEN ;
```

```
Screen # 24
( 24 SCAN1                                          JTR 11:53 03/07/88 )
: SCAN1  (    ---  . Scan Mode 1 command )
        OFF S.C.OUT           ( Off Scan Complete )
        STEP.PLAT.FID         ( Step Platform the fiducial )
        STEP.WAVE.FID         ( Step Wavelength to fiducial )
        BEGIN                 ( Start repetItive scans )
           ON S.C.OUT         ( Set Scan Complete )
           WAVE1 STEP.WAVE      ( Step to Wavelength 1 )
           SCAN20               ( Scan Platform 1/2 cycle )
           OFF S.C.OUT          ( Off Scan Complete )
           WAVE2 STEP.WAVE      ( Step to Wavelength 2 )
           SCAN20               ( Scan Platform 1/2 cycle )
           STEP.WAVE.FID        ( Step Wavelength to fiducial)
           STEP.PLAT.FID        ( Step Platform to fiducial)
           1 SCAN.PLAT.WAVE   ( Scan Platform and Wavelength)
        N.COM? UNTIL ;                ( Master version )


Screen # 25
( 25 SCAN2                                          JTR 10:11 02/05/88 )

: SCAN2 (  --  . Scan Mode 2 command )
        STEP.PLAT.FID           ( Step Platform to Fiducial )
        STEP.WAVE.FID           ( Step Wavelength to Fiducial )
        BEGIN
          1 SCAN.PLAT.WAVE        ( Scan Platform and Wavelength )
          N.COM? UNTIL            ( Check for new Command )
        ;


Screen # 26
( 26 SCAN3                                          JTR 10:12 02/05/88 )

: SCAN3 (  --  , Scan Mode 3 command )
        STEP.PLAT.FID         ( Set Platform to Fiducial )
        STEP.WAVE.FID         ( Step Wavelength to Fiducail )
        WAVE3  STEP.WAVE      ( Step Wavelength to Wave 3 )
        BEGIN                 ( Wait for new command )
          N.COM? UNTIL
        ;
```

```
Screen # 27
( 27 SUN? , SUN.CK                              JTR 16:23 03/05/88 )
: SUN.N.P@ (  -- F , Sun Not Present Bit 0-Alarm, 1- Safe )
       ANA@ 40 AND FLAG ;
: SUN?    (  -- F , Sun Present , 0-Safe, T-Alarm )
       SUN.P@  NOT DUP
          IF ELAP.T D@ SP.TO D! THEN    ( Set Time Out )
       SP.TO  D@  SP.D  0 D+  ELAP.T D@  D>
       OR ;

: SUN.CK    (  --   , Close Shutter if sun present )
       SUN?  IF
          NOOP      ( Action is automatic )
         ELSE
          RESET.S.N.P  ( Reset Sun Not Present )
       THEN ;



Screen # 28
( 28 H.V , NSP.OV , WT.OPT.T.I                  JTR 14:05 04/12/88 )
HEX
: H.V    (  F ---  , PTM High Voltage T-ON / F-auto )
       IF  C4@ 4 AND C4!    ( Sun Present Override Bit )
       ELSE  C4@ FB AND C4! THEN ;

: NSP.OV  (  ---  , Override Sun Present Command )
       ON SUN.P.OV.F  !    ( Set Sun Present Override Flag )
       ON H.V          ;    ( Set Sun Present Override Bit )

: WT.OPT.T.I  (    --    , Wait for Optical Test Indicator )
        BEGIN
           ANA@ 10 AND N.COM? OR  NOT WHILE
        SUN.CK REPEAT ;



Screen # 29
( 29 FCAL                                       JTR 10:37 02/05/88 )

: FCAL (  --  , Flight Calibration Command )
       ON T.LAMP          ( Test lamp on )
       SAVE.SYS           ( Remember system state )
       CLOSE  DUST        ( Close Dust Cover )
       ON   H.V           ( Force HV on )
       STEP.WAVE.FID      ( Step to Wavelength Fid )
       5 SCAN.WAVE        ( 5 Wavelength Scans )
       WT.OPT.T.I         ( Wait for Optical Test Indicator )
       5 SCAN.WAVE        ( 5 Wavelength Scans )
       OFF T.LAMP         ( Test Lamp Off )
       DUST.C @ DUST      ( Return Dust Cover )
       WAVE.C @ STEP.WAVE ; ( Return Wavelength )
```

```
Screen # 30
( 30 STAND.ST , OP.LAT                                  JTR 10:37 02/05/88 )

: STAND.ST  (  --   , Standard Start Configuration command )
        STEP.WAVE.FID      ( Step Wavelength to fiducial )
        S.DUST.O.FID       ( Open Dust cover )
        OPEN SHUT          ( Open Shutter )
        OPEN SHUT.C !
        OFF T.LAMP         ( Test Lamp Off )
        ON H.V             ( High voltage ON )
        ;

: OP.LAT  (  --  , Open Latch command )
        OPEN.LATCH            ( Open Latch )
        STEP.PLAT.FID ;       ( Step Platform to Fiducial )



Screen # 31
( 31 SHT.DW                                             JTR 10:37 02/05/88 )

: SHT.DW  (  --  , Normal Shut Down command )
        STEP.WAVE.FID         ( Step wavelength to fiducial )
        STEP.PLAT.FID         ( Step Platform to fiducial )
        OPEN SHUT             ( Open Shutter )
        OPEN SHUT.C !
        OFF T.LAMP            ( Test Lamp off )
        CLOSE.LATCH           ( Close latch )
        ;


        ( Turn off high voltage ???)



Screen # 32
( 32 SER.COM,                                           JTR 16:17 03/01/88 )
HEX
: SER.COM   (  --   , Service incoming commands )
    BEGIN  1 WHILE             ( Infinite loop )
        COM.CK                 ( Get new command )
            DUP 1 = IF SCAN1 THEN
            DUP 2 = IF SCAN2 THEN
            DUP 4 = IF SCAN3 THEN
            DUP 8 = IF NSP.OV THEN
            DUP 10 = IF FCAL THEN
            DUP 20 = IF STAND.ST THEN
            DUP 40 = IF OP.LAT THEN
                80 = IF SHT.DW THEN
        SUN?  STAND.ST
    REPEAT ;     ( Infinite loop )
```

```
Screen # 33
(                                                   JTR 10:36 02/05/88 )
```

```
Screen # 34
( 34 CT@                                            JTR 09:10 03/07/88 )
HEX
CODE CT@ ( -- N , Read Detector count, Port A3, B3   )
        0  A   MVI
        63     OUT        ( Stop Count )
        61     IN         ( Read Port A6, LSB )
        A  L   MOV        ( LSB )
        62     IN         ( Read Port B6 )
        A  H   MOV        ( MSB )
        2  A   MVI
        63     OUT        ( Reset Counter )
        1  A   MVI
        63     OUT        ( Enable Counter )
        HPUSH JMP  C: ( Place on stack )
```

```
Screen # 35
( 35 DATA.IN                                        JTR 09:16 03/07/88 )
: DATA.IN  ( --   , Read new Data, Store at DATA.BUF )
        CT@                 DATA.BUF !          ( Count )
        WAVE 03FF AND                           ( Wavelength )
        WAVE.FID? 400 AND +                     ( Wave Fid )
        SHUT@      800 AND +                    ( Shutter )
        HV@       1000 AND +                    ( High Voltage )
        T.LAMP@   2000 AND +                    ( Test Lamp )
        HV@       4000 AND +                    ( High Voltage )
        SUN.N.P@ 8000 AND + 2 DATA.BUF + ! ( Sun Not Present )
        PRES.COM 2000 * +                      ( Command )
        ERROR  F0 AND +    4 DATA.BUF + !    ( Error )
        ANG          1FFF AND                   ( Platform Angle )
        PLAT.FID?  2000 AND +                   ( Platform Fid )
        LATCH.FID? 4000 AND +                   ( Latch Status )
        ERR 8000   AND   + 5 DATA.BUF + ! ;     ( Error )
```

Laboratory Microsystems PC/FORTH 3.00   12:54   10/21/89   fom.scr

```
Screen # 36
( 36 DATA.OUT , Data handling in Forth      JTR 14:05 04/12/88 )

: DATA.OUT    (    --   , Data service on interrupt 7.5   )
        DATA# 0 DO
           BEGIN  O.DATA.C@ UNTIL       ( Wait for counter )
           DATA.BUF I  + C!  DATA.B.OUT   ( Output data byte )
         LOOP
        DATA.IN                     ( Read new Data   )
        DATA.BUF @ DATA.B.OUT      ( Preload output )
        ELAP.T D@ 1 0 D+ ELAP.T D!   ( Increment elapsed time )
        IRET ;                     ( End interrupt )
```

```
Screen # 37
( 37 INT7.5                                 JTR 15:12 02/19/88 )
HEX
CODE INT7.5   ( -- .   Set up Forth interrupt 7.5 )
        PSW PUSH B PUSH D PUSH H PUSH
        RSP  LHLD  H PUSH   ( Save Return Stack Pointer )
        ' DATA.OUT CFA H LXI     ( Address of word DATA.OUT )
        0072 JMP          ( NEXT1 . Mimic EXECUTE )
        C;
```

```
Screen # 38
( 38 Initiate Interrupt 7.5, INIT           JTR 12:58 01/05/89 )
HEX
        C3       2002 C!                 ( JMP command )
        '        INT7.5  2003 !   ( Upper interrupt Vector )

: INIT    (  --   , Initiate System )
        4000 DP !                    ( Set in flight HERE )
        DIS.I  I.VAR  I.PORTS          ( Initiate system )
        SUN.CK                       ( Check Sun Present )
        10000 MSEC  ." FP " CR   ( Wait 10 seconds, sign on )
        EI7.5&5.5 ( EI7.5 )          ( Enable Data )
        SER.COM ;                    ( Service incoming Commands )

    (   ' INIT  CFA  2000 ! ) ( Cold Start vector )
           ( **  Set Cold Start Vector before burning PROM ** )
 ( End of Flight Program )
```

```
Screen # 39
( Testing only                                     JTR 11:57 12/16/87 )
 HEX

 4000 HERE - DUP 64 2000 */
 DECIMAL CR SPACE . ." % " . ." Bytes remaining "  CR
```

```
Screen # 40
( LEAVE BLANK                                      JTR 15:58 08/10/87 )
```

```
Screen # 41
```

```
 0 ( FLIGHT PROGRAM  MASTER    JOB 612          JTR 12:54 01/05/89 )
 1 ( 1 USER VARIABLE                            JTR 12:57 01/05/89 )
 2 ( 2 CONSTANT                                 JTR 16:02 03/01/88 )
 3 ( 3 CONSTANT                                 JTR 15:58 01/06/89 )
 4 ( 4 1.PORTS                                  JTR 15:09 01/06/89 )
 5 ( 5 I.VAR                                    JTR 14:04 04/12/88 )
 6 ( 6 B2@, COM@ C4@                            JTR 15:39 02/19/88 )
 7 ( 7 DATA.B.OUT , C2@ , C2!                   JTR 15:44 02/19/88 )
 8 ( 8 C3! , MR! , MR@ , C4!                    JTR 15:41 02/19/88 )
 9 ( 9 FID@ , ANA@ , T.L@ , O.DATA.C@           JTR 15:29 02/03/88 )
10 ( 10 STEP.MOT , MOT.WAIT , WAVE.HS , RESET JTR 09:48 02/05/88 )
11 ( 11 PLAT.HS , DUST.HS , STEP.WAVE           JTR 09:51 02/05/88 )
12 ( 12 STEP.PLAT , STEP.DUST                   JTR 09:52 02/05/88 )
13 ( 13 LATCH.POW , LATCH.DIR , T.LAMP , SHUT JTR 16:11 03/05/88 )
14 ( 14 S.C.I@ , S.C.OUT , D.D.B.L , COM.RS     JTR 15:44 01/06/89 )
15 ( 15 WAVE.FID? , PLAT.FID? , DUST.O.FID?     JTR 15:59 01/06/89 )
16 ( 16 N.COM? , SEC                            JTR 16:13 03/05/88 )
17 ( 17 STEP.WAVE.FID , STEP.PLAT.FID           JTR 09:53 02/05/88 )
18 ( 18 S.DUST.O.FID , S.DUST.C.FID , DUST      JTR 10:24 02/05/88 )
19 ( 19 WAIT.DATA , CLOSE.LATCH , OPEN.LATCH    JTR 09:54 02/05/88 )
20 ( 20 SCAN.PLAT.WAVE                          JTR 14:05 04/12/88 )
21 ( 21 SCAN.WAVE                               JTR 10:11 02/05/88 )
22 ( 22 SCAN20 , SCAN.PLAT , SAVE.SYS           JTR 10:11 02/05/88 )
ok


23 (  23 COM.CK                                 JTR 16:28 03/01/88 )
24 (  24 SCAN1                                  JTR 11:33 03/07/88 )
25 (  25 SCAN2                                  JTR 10:11 02/05/88 )
26 (  26 SCAN3                                  JTR 10:12 02/05/88 )
27 (  27 SUN? , SUN.CK                          JTR 16:23 03/05/88 )
28 (  28 H.V , NSP.OV , WT.OPT.T.I              JTR 14:05 04/12/88 )
29 (  29 FCAL                                   JTR 10:37 02/05/88 )
30 (  30 STAND.ST , OP.LAT                      JTR 10:37 02/05/88 )
31 (  31 SHT.DW                                 JTR 10:37 02/05/88 )
32 (  32 SER.COM,                               JTR 16:17 03/01/88 )
33 (                                            JTR 10:36 02/05/88 )
34 (  34 CT@                                    JTR 09:10 03/07/88 )
35 (  35 DATA.IN                                JTR 09:16 03/07/88 )
36 (  36 DATA.OUT , Data handling in Forth      JTR 14:05 04/12/88 )
37 (  37 INT7.5                                 JTR 15:12 02/19/88 )
38 (  38 Initiate Interrupt 7.5, INIT           JTR 12:58 01/05/89 )
39 (  Testing only                              JTR 11:57 12/16/87 )
40 (  LEAVE BLANK                               JTR 15:58 08/10/87 )
41
ok
 ok
 ok
 ok
 ok
```

```
Screen # 0
( FLIGHT PROGRAM SECONDARY JOB 612          JTR 12:59 01/05/89 )
( Last change:   Screen  038                JTR 13:12 01/05/89 )


 ( APPENDIX XIII -   SECONDARY FLIGHT PROGRAM                      )
 (              Secondary unit without platform                    )
 (                                                                 )
 (                              SACS   Job 612                     )
 (                              8K PROM    UART                    )

 ( Version  1.0                                                    )
 ( J. T. Riley                                                     )
 ( RGI                                                             )


 ( File:  FPS.SCR        Disk:  HUP2   JOB 612  PROGRAM            )


Screen # 1
( 1 USER VARIABLE                           JTR 13:06 01/05/89 )
FORTH DEFINITIONS    HEX

35 USER ELAP.T            ( ELAPSED TIME, DP, Data periods )
3A USER PRES.COM          ( Present mode command )
3C USER WAVE              ( Wavelength motor step count )
3E USER WAVE.C            ( Wavelength currently commanded )
          ( Platform words removed )

44 USER H.V.C             ( High Voltage Status commanded )
46 USER DUST.C            ( Dust Cover position Commanded )
48 USER ERR               ( Error flag )
4A USER SUN.P.OV.F        ( Sun Present Override Flat T-Over )
4C USER SP.TO             ( Sun Present Time Out, DP )
4E USER SHUT.C            ( Shutter Position Commanded, T-closed)
                ( Secondary version )


Screen # 2
( 2 CONSTANT                                JTR 13:07 01/05/89 )
HEX
6000 CONSTANT DATA.BUF    ( Address of data buffer )
80FE CONSTANT RSP         ( Return Stack Pointer Address )
DECIMAL
 10  CONSTANT DATA.EN.R   ( Data Enable Rate, Hz )
  5  CONSTANT DATA#       ( Number of Data Bytes )
  7  CONSTANT DATA#.M     ( Number of Data Bytes of Master )
125  CONSTANT MOT.F       ( Stepper Motor Frequency, Hz )
 10  CONSTANT W.RELAY     ( Length relay pulse, msec )
400  CONSTANT TIME.OUT    ( Count limit for all motors, steps)
 10  CONSTANT MOT.DEL     ( Motor Delay , seconds )
        ( Platform words removed )
 40  CONSTANT BAUD        ( BAUD Rate constant, 40 - 4800 )
DATA.EN.R 60 * CONSTANT SP.D ( Sun Present Duration, Dat.En )
                    ( Secondary version )
```

```
Screen # 3
( 3 CONSTANT                                    JTR 15:58 01/06/89 )
DECIMAL

100   CONSTANT WAVE1      ( Step for 1356 Angstroms )    ( ???)
120   CONSTANT WAVE2      ( Step for 1383 Angstroms )    ( ???)
140   CONSTANT WAVE3      ( Step for 1551 Angstroms )    ( ???)
```

```
Screen # 4
( 4 1.PORTS                                     JTR 15:09 01/06/89 )
ASSEMBLER  HEX

CODE  I.PORTS     (  ---  . Initiate Ports )
        8 A MVI   44 OUT   ( Timer 2  /8 )
       C0 A MVI   45 OUT   ( Timer 2 cont Pulses )
       CD A MVI   40 OUT   ( A2-O, B2-I, C2-O )
       FF A MVI   64 OUT   ( Timer 3  /16384 )
       CF A MVI   65 OUT   ( Timer 3 cont Pulses )
       CD A MVI   60 OUT   ( A3-I, B3-I, C3-O )
     BAUD A MVI   84 OUT   ( Timer 4  /BAUD )
       40 A MVI   85 OUT   ( Timer 4 Sq wave )
       CE A MVI   80 OUT   ( A4-I, B4-O, C4-O )
       NEXT JMP C;
```

```
Screen # 5
( 5 I.VAR                                       JTR 14:07 04/12/88 )
HEX

: I.VAR   ( --  . Initialize Variables )
         0 0 ELAP.T D!        ( Zero Elapsed Time )
         SP.D @ 1+ 0 DMINUS  SP.TO D!  ( Sun Present Time Out )
         0 PRES.COM !     ( Present Command )
         0 ERR !          ( Error flag )
         0 H.V.C !        ( High Voltage Command )
         0 ERR !          ( Error Flag )
         0 SUN.P.OV.F ! ( Sun Present Override Flag )
         0 SHUT.C ! ;
```

```
Screen # 6
( 6 B2@, COM@, C4@                              JTR 15:39 02/19/88 )
HEX
CODE B2@ (  -- B. Read  Port B2 )
        42    IN       ( Read Port B2 )
        A  L  MOV      ( LSB )    0  H  MVI    ( MSB )
        HPUSH JMP  C; ( Place on stack )

CODE COM@ ( -- N , Read Command input, Port A4 )
        81    IN       ( Read Port A4)
        A  L  MOV      ( LSB )    0  H  MVI    ( MSB )
        HPUSH JMP  C; ( Place on stack )

CODE C4@ ( -- N , Read  Port C4 )
        83    IN       ( Read Port C4 )
        A  L  MOV      ( LSB )    0  H  MVI    ( MSB )
        HPUSH JMP  C; ( Place on stack )


Screen # 7
( 7 DATA.B.OUT , C2@ , C2!                      JTR 15:44 02/19/88 )
HEX
CODE DATA.B.OUT   ( B  -- . Data byte out. Port A2 )
        H    POP  L  A  MOV     ( Get From Stack )
        41 OUT   NEXT JMP  C; ( Out Port A2 )

CODE C2@ ( -- N . Read  Port C2 )
        41    IN       ( Read Port C2 )
        A  L  MOV      ( LSB )    0  H  MVI    ( MSB )
        HPUSH JMP  C; ( Place on stack )

CODE C2! ( N -- , Output Port C2 )
        H    POP  L  A  MOV     ( Get From Stack )
        41 OUT   NEXT JMP  C; ( Out Port C2 )


Screen # 8
( 8 C3! , MR! , MR@ , C4!                       JTR 15:41 02/19/88 )

CODE C3! ( N -- , Output Port C3 )
        H    POP  L  A  MOV     ( Get From Stack )
        63 OUT   NEXT JMP  C: ( Out Port C3 )
CODE MR! ( N -- , Output Motor Relay, Port B4 )
        H    POP  L  A  MOV     ( Get From Stack )
        82 OUT   NEXT JMP  C: ( Out Port B4 )
CODE MR@ ( -- N . Read  Port B4 )
        82    IN       ( Read Port B4 )
        A  L  MOV      ( LSB )    0  H  MVI    ( MSB )
        HPUSH JMP  C; ( Place on stack )

CODE C4! ( B -- , Output Port C4 )
        H    POP  L  A  MOV     ( Get From Stack )
        83 OUT   NEXT JMP  C; ( Out Port C4 )
```

```
Screen # 9
( 9 FID@ , ANA@ , T.L@ , O.DATA.C@            JTR 15:29 02/03/88 )
HEX

: FID@      ( -- N , Read Fiducials )
        C2@  2 OR  C2!      B2@ ;

: ANA@      (  -- N , Read Analog Levels )
        C2@  0FD AND  C2!   B2@ ;

: T.L@ ( -- N , Test Lamp Status, 0-Off, T-On )
        ANA@ 10 AND ;

: O.DATA.C@ ( -- N , Output Data Counter, 0-not/8, 20-/8 )
        ANA@ 20 AND ;



Screen # 10
( 10 STEP.MOT , MOT.WAIT , WAVE.HS , RESET JTR 09:48 02/05/88 )
HEX
: STEP.MOT   ( N --   , Step motor N once )
        C3!  1 MSEC 0 C3! ;

: MOT.WAIT   ( --   , Wait one motor cycle )
        1 MOT.F /  1000 * 1-   MSEC ;

: WAVE.HS     ( F --  , Wavelength motor, 0-Hold, 1-Step)
        1 AND  1+ MR!
        W.RELAY MSEC 0 MR! ;

: RESET.S.N.P  (  ---    ,  Reset Sun Not Present )
        C4@ DUP 8 AND C4! 10 MSEC  ( Pulse command )
        C4! ;



Screen # 11
( 11 DUST.HS , STEP.WAVE                      JTR 13:07 01/05/89 )
HEX
        ( Platform word removed )


: DUST.HS     ( F --  , Dust Cover motor, 0-Hold. 1-Step)
        4 AND  4 + MR!
        W.RELAY MSEC 0 MR! ;

: STEP.WAVE    ( N --   , Step Wavelength N steps )
        ON WAVE.HS
        0 DO 4 STEP.MOT MOT.WAIT
        LOOP
        OFF WAVE.HS ;

                ( Secondary version )
```

```
Screen # 12
( 12 STEP.DUST                                   JTR 13:07 01/05/89 )

        ( Platform word removed )

: STEP.DUST    ( N --   , Step Dust Cover motor N steps )
        ON DUST.HS
        0 DO   8 STEP.MOT   MOT.WAIT
        LOOP
        OFF DUST.HS ;



                ( Secondary version )


Screen # 13
( 13 T.LAMP , T.LAMP@ , SHUT , SHUT@          JTR 13:08 01/05/89 )

: T.LAMP    ( F -- , Test Lamp Command. 0-Off. T-ON )
        1 AND C4@ FE AND +   C4! ;
: T.LAMP@  (    -- F , Test Lamp Indicator, 0-Off. T-On )
        ANA@ 10 AND FLAG ;
: SHUT   ( F -- , Shutter Command, 0-Open, T-Closed )
        2 AND C4@ FD AND +   C4! ;

: SHUT@ (    -- N , State of Shutter. 0-open, 2-closed)
        C4@ 2 AND ;



            ( Secondary version )


Screen # 14
( 14 S.C.I@ , S.C.OUT , D.O.B.L , COM.RS   JTR 15:44 01/06/89 )

: S.C.I@ ( -- F , Scan Complete ;Input, 0-No, T-Complete )
        ANA@ 80 AND ;

: S.C.OUT  ( F -- , Scan Complete Out, 0-Not, T-Complete )
        10 AND C4@ EF AND +   C4! ;

: D.O.B.L  (  -- , Data Out Buffer Load, Pulse )
        4  C4@  +  C4!     ( Set bit )
        C4@ FB AND C4! ;  ( Clear Bit )

: COM.RS   (  -- , Command Reset )
        20  C4@  +  C4!     ( Set bit )
        C4@ DF AND C4! ;  ( Clear Bit )
```

```
Screen # 15
( 15 WAVE.FID? , DUST.O.FID? , DUST.C.FID? JTR 13:08 01/05/89 )
HEX
: WAVE.FID?    (  -- F . At Wavelength Fid, T-at fid )
       FID@ 3 AND FLAG ;    ( Two Fids )


: DUST.O.FID?    (  -- F . At Dust Cover Open Fid, T-Open)
       FID@  4 AND FLAG  ;
: DUST.C.FID?    (  -- F , At Dust Cover Closed Fid, T-Closed)
       FID@  8 AND FLAG  ;

            ( Platform words removed )


               ( Secondary version )



Screen # 16
( 16 N.COM? , SEC                              JTR 16:13 03/05/88 )

: N.COM?    (  -- F,  New Command ? , T-NEW )
       COM@
       PRES.COM @ =  NOT ;  ( Does not change command )

: SEC       (  N  --  , Wait Seconds )
       0 DO ELAP.T D@ DATA.EN.R @ 0 D+
             BEGIN
                2DUP ELAP.T D@ D= N.COM? OR UNTIL
       DROP DROP LOOP ;

: HV@  (  --- F , High Voltage Status 0-low, T- Good )
       ANA@ 2 AND FLAG ;



Screen # 17
( 17 STEP.WAVE.FID                             JTR 13:08 01/05/89 )

: STEP.WAVE.FID   ( --  . Step to Wavelength Fiducial )
       TIME.OUT
       BEGIN 1- DUP  WAVE.FID? NOT AND  WHILE  ( At fid ? )
          1 STEP.WAVE REPEAT DROP ;    ( Step to fid )

       ( Platform word removed )




              ( Secondary version )
```

```
Screen # 18
( 18 S.DUST.O.FID , S.DUST.C.FID , DUST     JTR 13:08 01/05/89 )

: S.DUST.O.FID (  --  , Open Dust Cover with time out )
        TIME.OUT
        BEGIN 1- DUP  DUST.O.FID? NOT AND WHILE
           1 STEP.DUST REPEAT DROP ;     ( Step to fid )

: S.DUST.C.FID (  --  , CLOSE Dust Cover with time out )
        TIME.OUT
        BEGIN 1- DUP  DUST.C.FID? NOT AND WHILE
           1 STEP.DUST REPEAT DROP ;     ( Step to fid )

: DUST    ( F --  , Open or Close Dust Cover, T-Open )
        IF  BEGIN S.C.I@ NOT UNTIL  S.DUST.O.FID
        ELSE  S.DUST.C.FID THEN ;
                 ( Secondary version )


Screen # 19
( 19 WAIT.DATA                               JTR 13:09 01/05/89 )

: WAIT.DATA (    --  Wait for one data out )
        ELAP.T @ 1+
        BEGIN DUP ELAP.T @ = UNTIL DROP ;

        ( Platform words removed )




                 ( Secondary version )


Screen # 20
(                                            JTR 13:09 01/05/89 )


  (  Platform word removed )







                ( Secondary version )
```

```
Screen # 21
( 21 SCAN.WAVE                                   JTR 10:11 02/05/88 )

: SCAN.WAVE  ( N --  , Scan in Wavelength only )
        0 DO                    ( Multiple Scans )
           TIME.OUT
           BEGIN 1- DUP  WAVE.FID? NOT AND WHILE
                1 STEP.WAVE   WAIT.DATA  ( Wait for Data Out )
           REPEAT DROP          ( Next Wavelength step )
        LOOP ;                  ( Next Scan )




Screen # 22
( 22 SAVE.SYS                                    JTR 13:09 01/05/89 )


        ( Platform words removed )


: SAVE.SYS  (   --   , Save System state )
        WAVE @ WAVE.C !   ANG @ ANG.C ! ;



                ( Secondary version )


Screen # 23
( 23 COM.CK                                      JTR 16:28 03/01/88 )

: COM.CK  (  -- N , Check for New Command )
        N.COM?  DUP  IF
           10 SEC
              N.COM? IF
           DROP COM@ DUP PRES.COM ! D.O.B.L
        THEN THEN ;
```

```
Screen # 24
( 24 SCAN1                                          JTR 13:09 01/05/89 )

: SCAN1  (   ---  , Scan Mode 1 command )
        STEP.WAVE.FID        ( Step Wavelength to fiducial )
        BEGIN                ( Start repetItive scans )
           BEGIN S.C.I@ NOT UNTIL ( Wait for Scan Complete )
           OFF S.C.OUT       ( Reset S.C.OUT )
           WAVE1 STEP.WAVE    ( Step to Wavelength 1 )
           WAVE2 STEP.WAVE    ( Step to Wavelength 2 )
           STEP.WAVE.FID      ( Step Wavelength to fiducial)
           1 SCAN.WAVE        ( Scan Platform and Wavelength)
           ON S.C.OUT        ( Set S.C.Out )
        N.COM? UNTIL ;    ( Check for New command )

           ( Secondary version )


Screen # 25
( 25 SCAN2                                          JTR 13:09 01/05/89 )

: SCAN2 (  --  , Scan Mode 2 command )
        STEP.WAVE.FID          ( Step Wavelength to Fiducial )
        BEGIN
           BEGIN S.C.I@ NOT UNTIL ( Wait for Scan Complete )
           OFF S.C.OUT       ( Reset S.C.OUT )
          1 SCAN.WAVE         ( Scan Wavelength )
           ON S.C.OUT        ( Set S.C.Out )
          N.COM? UNTIL          ( Check for new Command )
        ;

           ( Secondary version )


Screen # 26
( 26 SCAN3                                          JTR 13:09 01/05/89 )

: SCAN3 (  --  , Scan Mode 3 command )
        ON S.C.OUT       ( Set S.C.Out )
        BEGIN S.C.I@ NOT UNTIL ( Wait for Scan Complete )
        OFF S.C.OUT      ( Reset S.C.OUT )
        STEP.WAVE.FID        ( Step Wavelength to Fiducial )
        WAVE3  STEP.WAVE     ( Step Wavelength to Wave 3 )
        BEGIN                ( Wait for new command )
          N.COM? UNTIL
        ;

        ( Secondary version )
```

```
Screen # 27
( 27 SUN? , SUN.CK                          JTR 16:23 03/05/88 )
: SUN.N.P@ (  -- F , Sun Not Present Bit 0-Alarm, 1- Safe )
        ANA@ 40 AND FLAG ;
: SUN?    (  -- F , Sun Present , 0-Safe, T-Alarm )
        SUN.P@   NOT DUP
           IF ELAP.T D@ SP.TO D! THEN    ( Set Time Out )
        SP.TO  D@  SP.D  0 D+  ELAP.T D@  D>
        OR ;

: SUN.CK    (  --  , Close Shutter if sun present )
        SUN?  IF
           NOOP      ( Action is automatic )
          ELSE
           RESET.S.N.P  ( Reset Sun Not Present )
        THEN ;
```

```
Screen # 28
( 28 H.V , NSP.OV , WT.OPT.T.I                JTR 14:07 04/12/88 )
HEX
: H.V    (  F ---  , PTM High Voltage T-ON / F-auto )
        IF  C4@ 4 AND C4!   ( Sun Present Override Bit )
        ELSE  C4@ FB AND C4! THEN ;

: NSP.OV  (  ---  . Override Sun Present Command )
        ON SUN.P.OV.F  !    ( Set Sun Present Override Flag )
        ON H.V         ;    ( Set Sun Present Override Bit )

: WT.OPT.T.I  (   --   , Wait for Optical Test Indicator )
         BEGIN
            ANA@ 10 AND N.COM? OR  NOT WHILE
         SUN.CK REPEAT ;
```

```
Screen # 29
( 29 FCAL                                   JTR 10:37 02/05/88 )

: FCAL (  --  , Flight Calibration Command )
        ON T.LAMP          ( Test lamp on )
        SAVE.SYS           ( Remember system state )
        CLOSE  DUST        ( Close Dust Cover )
        ON   H.V           ( Force HV on )
        STEP.WAVE.FID      ( Step to Wavelength Fid )
        5 SCAN.WAVE        ( 5 Wavelength Scans )
        WT.OPT.T.I         ( Wait for Optical Test Indicator )
        5 SCAN.WAVE        ( 5 Wavelength Scans )
        OFF T.LAMP         ( Test Lamp Off )
        DUST.C @ DUST      ( Return Dust Cover )
        WAVE.C @ STEP.WAVE ; ( Return Wavelength )
```

```
Screen # 30
( 30 STAND.ST                                   JTR 13:10 01/05/89 )

: STAND.ST  (  --  , Standard Start Configuration command )
        STEP.WAVE.FID     ( Step Wavelength to fiducial )
        S.DUST.O.FID      ( Open Dust cover )
        OPEN SHUT         ( Open Shutter )
        OPEN SHUT.C !
        OFF T.LAMP        ( Test Lamp Off )
        ON H.V            ( High voltage ON )
        ;

           ( Platform word removed )


                ( Secondary version )



Screen # 31
( 31 SHT.DW                                     JTR 14:46 03/05/88 )

: SHT.DW  (  --  , Normal Shut Down command )
        STEP.WAVE.FID        ( Step wavelength to fiducial )
        OPEN SHUT            ( Open Shutter )
        OPEN SHUT.C !
        OFF T.LAMP           ( Test Lamp off )
        CLOSE.LATCH          ( Close latch )
        ;


        ( Turn off high voltage ???)




Screen # 32
( 32 SER.COM.                                   JTR 13:10 01/05/89 )
HEX
: SER.COM   (  --  , Service incoming commands )
    BEGIN  1 WHILE            ( Infinite loop )
        COM.CK                ( Get new command )
            DUP 1 = IF SCAN1 THEN
            DUP 2 = IF SCAN2 THEN
            DUP 4 = IF SCAN3 THEN
            DUP 8 = IF NSP.OV THEN
            DUP 10 = IF FCAL THEN
            DUP 20 = IF STAND.ST THEN
                80 = IF SHT.DW THEN
        SUN?  STAND.ST
    REPEAT ;     ( Infinite loop )

                ( Secondary version )

Laboratory Microsystems PC/FORTH 3.00      13:01   10/21/89    fos.scr
```

```
Screen # 33
(                                              JTR 10:36 02/05/88 )
```

```
Screen # 34
( 34 CT@                                       JTR 14:52 03/05/88 )
HEX
CODE CT@ ( -- N , Read Detector count. Port A3, B3  )
        0  A  MVI
        63    OUT       ( Stop Count )
        61    IN        ( Read Port A6, LSB )
        A  L  MOV       ( LSB )
        62    IN        ( Read Port B6 )
        A  H  MOV       ( MSB )
        2  A  MVI
        63    OUT       ( Reset Counter )
        1  A  MVI
        63    OUT       ( Enable Counter )
        HPUSH JMP  C; ( Place on stack )
```

```
Screen # 35
( 35 DATA.IN                                   JTR 13:10 01/05/89 )
: DATA.IN   ( --  , Read new Data, Store at DATA.BUF )
        CT@                  DATA.BUF !        ( Count )
        WAVE 03FF AND                          ( Wavelength )
        WAVE.FID? 400 AND +                    ( Wave Fid )
        SHUT@      800 AND +                   ( Shutter )
        HV@       1000 AND +                   ( High Voltage )
        T.LAMP@   2000 AND +                   ( Test Lamp )
        HV@       4000 AND +                   ( High Voltage )
        SUN.N.P@ 8000 AND + 2 DATA.BUF + !  ( Sun Not Present )
        PRES.COM 2000 * +                       ( Command )
        ERROR  F0 AND +   4 DATA.BUF + !       ( Error )
                 ;

             ( Secondary version )
```

```
Screen # 36
( 36 DATA.OUT . Data handling in Forth      JTR 13:10 01/05/89 )

: DATA.OUT    (   --  . Data service on interrupt 7.5   )
        DATA#.M     0 DO
           BEGIN  S.DATA.C@ UNTIL ( Wait Master Data )
         LOOP
        DATA# 0 DO
           BEGIN  S.DATA.C@ UNTIL        ( Wait for counter )
           DATA.BUF I  + C@  DATA.B.OUT  ( Output data byte )
         LOOP
        DATA.IN                    ( Read new Data   )
        DATA.BUF @ DATA.B.OUT     ( Preload output )
        ELAP.T D@ 1 0 D+ ELAP.T D!   ( Increment elapsed time )
        IRET ;                   ( End interrupt )

                            ( Secondary version )


Screen # 37
( 37 INT7.5                                      JTR 15:12 02/13/88 )
HEX
CODE INT7.5   ( -- .  Set up Forth interrupt 7.5 )
        PSW PUSH B PUSH D PUSH H PUSH
        RSP  LHLD  H PUSH  ( Save Return Stack Pointer )
        ' DATA.OUT CFA H LXI     ( Address of word DATA.OUT )
        2072 JMP           ( NEXT1 . Mimic EXECUTE )
        C;



Screen # 38
( 38 Initiate Interrupt 7.5, INIT             JTR 13:12 01/05/89 )
HEX
        C3       2002 C!                ( JMP command )
        '        INT7.5  2003 !    ( Upper interrupt Vector )

: INIT    (  --  . Initiate System )
        4200 DP !                  ( Set in flight HERE )
        DIS.I  I.VAR  I.PORTS        ( Initiate system )
        GUN.OK                     ( Check Gun Present )
        10000 MSEC  ." FP " CR   ( Wait 10 seconds, sign on )
        EI7.5&5.5 ( EI7.5 )        ( Enable Data )
        SER.COM ;                  ( Service incoming Commands )

    (   ' INIT  CFA  2000 ! ) ( Cold Start vector )
           ( **  Set Cold Start Vector before burning PROM ** )
 ( End of Secondary Flight Program )


Laboratory Microsystems PC/FORTH 3.00        13:00  02/21/89   fos.scr
```

```
Screen # 39
( Testing only                                    JTR 11:57 12/16/87 )
 HEX

 4000 HERE - DUP 64 2000 */
 DECIMAL CR SPACE . ." % " . ." Bytes remaining "  CR
```

```
Screen # 40
( LEAVE BLANK                                     JTR 15:58 08/10/87 )
```

```
Screen # 41
```

```
  0 ( FLIGHT PROGRAM SECONDARY JOB 612              JTR 12:54 01/05/89 )
  1 ( 1 USER VARIABLE                               JTR 13:06 01/05/89 )
  2 ( 2 CONSTANT                                    JTR 13:07 01/05/89 )
  3 ( 3 CONSTANT                                    JTR 15:58 01/06/89 )
  4 ( 4 1.PORTS                                     JTR 15:09 01/06/89 )
  5 ( 5 I.VAR                                       JTR 14:07 04/12/88 )
  6 ( 6 B2@, COM@ C4@                               JTR 15:39 02/19/88 )
  7 ( 7 DATA.B.OUT , C2@ , C2!                      JTR 15:44 02/19/88 )
  8 ( 8 C3! , MR! , MR@ , C4!                       JTR 15:41 02/19/88 )
  9 ( 9 FID@ , ANA@ , T.L@ , O.DATA.C@              JTR 15:29 02/03/88 )
 10 ( 10 STEP.MOT , MOT.WAIT , WAVE.HS , RESET JTR 09:48 02/05/88 )
 11 ( 11 DUST.HS , STEP.WAVE                        JTR 13:07 01/05/89 )
 12 ( 12 STEP.DUST                                  JTR 13:07 01/05/89 )
 13 ( 13 T.LAMP , T.LAMP@ , SHUT , SHUT@            JTR 13:08 01/05/89 )
 14 ( 14 S.C.I@ , S.C.OUT , D.O.B.L , COM.RS        JTR 15:44 01/06/89 )
 15 ( 15 WAVE.FID? , DUST.O.FID? , DUST.C.FID? JTR 13:08 01/05/89 )
 16 ( 16 N.COM? , SEC                               JTR 16:13 03/05/88 )
 17 ( 17 STEP.WAVE.FID                              JTR 13:08 01/05/89 )
 18 ( 18 S.DUST.O.FID , S.DUST.C.FID , DUST         JTR 13:08 01/05/89 )
 19 ( 19 WAIT.DATA                                  JTR 13:09 01/05/89 )
 20 (                                               JTR 13:09 01/05/89 )
 21 ( 21 SCAN.WAVE                                  JTR 10:11 02/05/88 )
 22 ( 22 SAVE.SYS                                   JTR 13:09 01/05/89 )
ok


 23 (  23 COM.CK                                    JTR 16:28 03/01/88 )
 24 (  24 SCAN1                                     JTR 13:09 01/05/89 )
 25 (  25 SCAN2                                     JTR 13:09 01/05/89 )
 26 (  26 SCAN3                                     JTR 13:09 01/05/89 )
 27 (  27 SUN? , SUN.CK                             JTR 16:23 03/05/88 )
 28 (  28 H.V , NSP.OV , WT.OPT.T.I                 JTR 14:07 04/12/88 )
 29 (  29 FCAL                                      JTR 10:37 02/05/88 )
 30 (  30 STAND.ST                                  JTR 13:10 01/05/89 )
 31 (  31 SHT.DW                                    JTR 14:46 03/05/88 )
 32 (  32 SER.COM,                                  JTR 13:10 13/05/89 )
 33 (                                               JTR 10:36 02/05/88 )
 34 (  34 CT@                                       JTR 14:52 03/05/88 )
 35 (  35 DATA.IN                                   JTR 13:10 01/05/89 )
 36 (  36 DATA.OUT , Data handling in Forth         JTR 13:10 01/05/89 )
 37 (  37 INT7.5                                    JTR 15:12 02/19/88 )
 38 (  38 Initiate Interrupt 7.5, INIT              JTR 13:12 01/05/89 )
 39 (  Testing only                                 JTR 11:57 12/16/87 )
 40 (  LEAVE BLANK                                  JTR 15:58 08/10/87 )
 41
ok
 ok
 ok
 ok
 ok
```

APPENDIX G

RESEARCH SUPPORT INSTRUMENTS, INC.

CALIBRATION AND TESTING SYSTEM

USING THE IBM PC

Research Support Instruments, Inc.
10610 Beaver Dam Road
Cockeysville, Maryland 21030
(301) 785-6250

J. T. Riley
01/05/89
File: CTS612.DOC
CTSCUS.DOC
Illustrations
CTSAPP.DOC

TABLE OF CONTENTS

ILLUSTRATIONS

APPENDIX

ASSOCIATED PRINTS

IBM SPACECRAFT EMULATOR CARD SCHEMATIC D910-224-0-5

CONTACT PERSONS

AFGL

Charles Forsberg (612) 377-2625

RSI

Tom Riley (301) 785-6250

IBM PC CALIBRATION AND TESTING SYSTEM

Introduction

The purpose of this doucment is described. The components and capabilities of the CTS system are listed.

1.0 Purpose

This document describes the RSI Calibration and Testing System (CTS). Included are descriptions of the RSI CTS card, the complete system hardware, and the demonstration software.

The RSI CTS System allows a standard IBM PC or compatible microcomputer to be used as the CTS for scientific instruments bound for space flight. This system enables the scientist to completely check out the system before flight, to calibrate the flight instrument, and to recheck the instrument after the flight.

On large complicated instruments, the IBM based CTS may be used to test the individual components before integration into the completed experiment. This builds confidence in the pieces before difficult tests of a complete system are run. On large systems having a number of data channels, the individual IBM based CTS's can be used for each channel. The individual IBM's may then be combined in a Local Area Network (LAN). An additional IBM can then be used as the network controller and to store the data on large tape.

For instruments which incorporate their own microprocessors, the IBM PC used for the CTS may also be used for software development. This will require additional commercial software (Assembler, Forth, etc.) and hardware (additional cables and a PROM Burner). Using the same instrument for the CTS and software development can result in substantial savings.

1.1 System Components

The RSI CTS System (see Figure 22) consists of:

* 1. IBM PC microcomputer or compatible with
  Multifunction Card
  (Mem, RS-232, RT Clock, Printer Port)
  Data Storage Medium (2nd disk, hard disk, or tape)
2. RSI CTS Card
* 3. +28 Volt Bench Power Supply
4. Cables
5. Software Package
6. Documents

* These items are not provided under this contract.

1.2 Capabilities

The RSI CTS Hardware will provide the following support:

1. Instrument Power (+28 Volt)
2. Spacecraft Command Signals (8)
3. Instrument Status Signals (8)
4. Data Enable Signal
5. Data Clock Signal (gated and ungated)
6. Display of Text and Graphics
7. Data Storage

This system with supporting software will allow the user to:

1. Display Test Data in real time
2. Issue commands to the flight instrument
3. Set Data Clock Rates, Enable timing, and number of data bits.
4. Store data in RAM in real time
5. Store Blocks of Data on permanent medium.
6. Add time, date, and headings to data blocks automatically
7. Print out data and test screens.
8. Automatically run timed sequences of tests.
9. Development of new testing sequences
10. Add to or develop new CTS software

HARDWARE

Introduction

The hardware for the CTS system is detailed.

2.0 Hardware

This system uses the following hardware.

2.1 Microcomputer

The CTS computer must meet the following requirements.

2.1.1 Computer

The CTS computer may be any system which can run programs in PC/FORTH, store data in IBM PC formats, and accept IBM PC cards. The choice is a question of price as much as anything else. Most PC clones will meet these requirements. You may, however, wish to run this demonstration program on the system with the RSI CTS card and a multifuction card installed before you buy the computer.

You should also carefully consider whether you will need a transportable system and the type of permanent data storage you wish.

The 8087 Math co-processor is not necessary and is not helpful for the CTS software. If you will also be using the IBM PC for data reduction or graphics, then the co-processor would be helpful.

An IBM AT clone with a faster clock, 10 MHz instead of 4.77 MHz, would speedup the program and allow faster data taking and storage. The interrupt system for the AT is, however, slightly different from the PC. This may cause a problem in data taking unless the version of Forth used is specifically adapted for the AT. The CTS hardware and software should be tested on an AT before purchase.

2.1.2 Multifunction Card

The multifunction card should include at least 360K of additional RAM, a real time clock, and a printer port. A RS-232 port may also be needed if you plan to do flight software development.

2.1.3 Mass Media

The permanent data medium may be any form of mass storage available for the IBM PC. The more data storage required, the higher the price. The cheapest is the standard floppy drive. These hold about 300K bytes of data and are adequate for calibration tests at modest data rates and bit counts.

High density floppy drives will go to about 1M byte and will cover most ground calibration requirements. Hard disks come in 10's of megabytes but may not transport as well and should be backed up on other media. Cartridge tape systems are available both as independent units and as back-ups for disk drives. They hold up to 60 megabytes and are popular for large data storage.

2.2 RSI CTS Card

This card allows an IBM PC to emulate the spacecraft. It can provide all signals which in flight go from the spacecraft to the instrument and can accept all signals which go from the instrument to the spacecraft.

2.2.1 Command Signals

Eight (8) command signals are available which can provide either voltage levels or pulses to the instrument. The voltage level for the output is set by applying power from an external power supply and may be any value from 5 to 40 volts (28 V being most common). A temporary shunt on the RSI CTS card (Shunt 4) will allow the +5 V to power these outputs for simple tests.

In the Get Away Special (GAS) system for the shuttle, for example, these command lines mimic the relay closures in the GAS Command Decoder (GCD) module. The GCD responds to commands sent by the Mission Specialist and usually contains 3 double throw single pole relays. More complicated instruments might have more than one GCD.

The command lines may be used either as eight (8) separate command lines or interpreted as an 8 bit number. This raises the number of possible commands to 256 but also increases the possiblity of getting the command wrong. Small changes in the CTS software will accommodate either system.

2.2.2 Status Signals

Eight (8) digital status lines may be read by the computer. These usually show instrument conditions such as alarm conditions or the presence of voltages within the flight instrument. These signals usually change slowly and these lines have resistor capacitor filters.

In the GAS system these status lines would be sent back to the Mission Specialist through the Status Responder Unit

(SRU). They allow the Mission Specialist to see if the instrument is running properly.

In larger systems, which are connected to ground through telemetry, a few of these lines may be paired with a few command lines to provide a serial data channel. This requires at least one data line and one handshake line in each direction. This system is slow and requires alot of checks, but it can be used to make setting changes on the flight instrument during flight.

2.2.3 Data Clock and Enable

This card generates Data Clock and Data Enable signals to transfer data from the flight instrument to CTS. This process mimics the transfer of data from the flight instrument to flight data storage or telemetry systems.

The Data Clock rate, the Data Enable rate, and the number of Data Bits are all software setable. All these signals have 0 at 0 volts and 1 at +5 volts (TTL compatible).

In the most common arrangement, the flight instrument contains a shift register for the data. This register is loaded with the first word (8 or 16 bits) before the Enable signal occurs. When the Enable signal does occur, it enables the operation of the shift register and, if the flight insturment is microprcessor controlled, triggers an interrupt. The CTS Card reads the first bit of data just after the Data Enable signal, the first bit therefore should be waiting on the output line.

One half cycle later the Data Clock shifts the shift resister and the second data bit appears on the output line. The data bits are read into the CTS memory and shifted out of the flight instrument on alternate half cycles from the Data Clock. This insures that the data will be stable when it is read. The flight instrument is responsible for detecting when the shift register must be refilled and this process must be completed in less than one half of a clock cycle. This detection and refilling process often sets the maximum clock speed the flight instrument can handle.

The timing of the Data Clock and the Data Enable is illustrated in Appendix XV

The Gated Data Clock provides one cycle for every bit of data transferred. A Continuous Data Clock is also available on the output connector. The Data Rate of this clock is setable in software from 1M to 40 Hz but the range from 128 KHz to 5 KHz is most often used. Rates over 128KHz are too fast for most flight instrument microprocessors and may have transmission problems. Very slow rates simply do not transfer much data. Stopping the Data Rate with software also stops the Data Enable signal.

The Data Enable signal may be set to occur as often as once every 16 Data Clock cycles or as rarely as once every 131072 Data Clock cycles. The normal range is from 100Hz to 1Hz. The Data Enable signal is an envelope for the Gated Data Clock.

The Number of Data Bits is also software setable and is the number of data bits in the gated clock. The number of bits may be set from 6 to 256. The number is usually a multiple of 8. These limits are set by the First In First Out (FIFO) memory on the card.

The Data Clock rate and the Data Enable rate may be changed from software during a test. If the number of Data Bits is changed during a test, however, the display of data on the IBM monitor will be disrupted. The number of bits may be changed along with the display for testing different flight instruments.

2.2.4 Data In

Data from the flight instrument is stored in an 8 by 32 byte (256 bit) FIFO which can then be read into main memory by software. This is done at the end of each Data Enable signal.

2.2.5 Address

The address of the IBM CTS card may be set by hardware DIP switch on the card to 16 different I/O address blocks. It is set at the factory to 300H to 31FH which is assigned to the Protype Card. If you are using other prototype cards you may wish to move the starting address to one of the others listed in Appendix XIV. The CTS.ADDR in Screen #2 of the software listing will also have to be changed. This is the only software change for a change in address.

2.2.6 Interrupt

At the end of the Data Enable signal a hardware interrupt signal may be sent to the IBM PC. This interrupt means that new data is ready to be read in.

A hardware shunt (Shunt 5) on the card allows the interrupt to be set to any of four (4) interrupt request lines (IRQ-2, IRQ-3, IRQ-4, IRQ-5) which are described in Appendix XIV. The card is factory set to IRQ-3 which is normally used for the Secondary Asynchronous Communications Port (COM2).

Changing the INTERRRUPT# constant in Screen #2 of the software is the only software change required.

To facilitate testing of the CTS card a separate shunt (Shunt 3) is provided to allow the interrupt to be hardware disabled. With this shunt removed the interrupt software may be completely debugged before you must deal with the actual hardware interrupt.

The AT computer uses a slightly different Interrupt system but the PC system used on the CTS card will work properly if the interrupt used by the card is not used by for anything else and the software language used is designed for use on the AT.

2.3 Power Supply

No power is taken from the microcomputer for hardware outside the computer. A separate power supply is provided +28 volt power to run the instrument. This power supply runs on 115 Volts AC and the power to the flight instrument has a separate toggle switch.

The power may be provided to the instrument over a separate power cable or this cable may be combined with the data cable.

This power may also be used to power the Command lines if you wish them to be at the 28 Volt level.

2.4 Commercial Cards

You may require hardware features not provided by the CTS card. These can be provided by commercially available cards for the IBM PC. The flexible addressing and interrupt selection on the CTS card makes coordinating with other cards much easier.

One common need is to read analog housekeeping lines such as temperature, or voltage monitors. These are usually single ended, have a range of about 0 to 5 volts, and require only 8 bit accuracy.

One card that serves this purpose well is the 83-064A Universal I/O card from

John Bell Engineering, Inc.
400 Oxford Way
Belmont, CA 94002

It has 16 single ended 8 bit analog inputs and 9 eight bit digital ports. This card costs about $300, and you will probably need to make up an analog cable. This card is mapped into the high software interrupt addresses and does not interfere with the CTS card.

2.5 Cables

The flight instrument is usually stored in a clean area with limited access. The CTS may, therefore, operate over cables of up to 50 feet. The Data Clock, Data Enable, and Data Signals must be on either coax or twisted shielded pairs. The Command lines, power, and return lines may be on unshielded wires.

Some flight instruments may require clock and data signals at levels other than TTL. CMOS levels (Ø and +1Ø) are common as well as inverted and differential signals. The commands can be set to any desired level by providing external power and can be inverted by simply changing software. If other signals are needed at non-TTL levels then a small hand wired circuit board will be needed to change signal levels. This device is usually associated with bench power supply and powered by it.

Often the flight instrument must be placed in a vacuum chamber for ground calibration. If this is the case, a second set of cables may be needed to match the connectors on the chamber.

GENERAL PURPOSE SOFTWARE

Introduction

The general requirements for the CTS software are discussed.

3.0 General Purpose Software

The software described in this section applies to all applications using the RSI CTS card. This specific application is described in the next section.

3.1 Forth

The CTS software is written in PC/FORTH from :

Laboratory Microsystems, Inc.
P.O. Box 10430
Marina del Rey, CA 90295

A copy of this computer language will be needed if you wish to rewrite, or modify the provided software. On the disk provided is a autostart version of the demonstration software and a screen version. The autostart version can be run from IBM DOS simply by calling its name. The screen version is the same but requires the FORTH language to run. The screen version is open for modification by the FORTH editor.

LMI also makes a graphics extension for PC/FORTH.

PC/FORTH follows the 83-Standard for Forth. Other versions of Forth on the IBM PC will run the demonstration software but small modifications to the program may be needed.

3.2 CTS Software

The program provided, CTS.SCR, is intended as a demonstration of the use of the RSI CTS system. It is heavily documented and can serve as a base for the development of your CTS software.

3.2.1 Initilization

This section sets up the computer and RSI CTS card. If you change the card's address or interrupt you need only change numbers in Screen #2 of this section.

A major part of this section is the assembly language driver words which allow communication with the RSI CTS card. Having these words available will save you a lot of time in developing your CTS program.

3.2.2 Testing from the Keyboard

This section of the program allows you to run tests from the keyboard of the IBM PC. You can start and stop the clocks, take data, and issue commands. This allows you to fully check out your flight instrument.

3.2.3 Automatic Testing

This feature lets you set up a timed sequence of commands. This allows you to repeat a complex test in exactly the same way, time after time. Such a test might be used in calibration tests or before and after the flight tests.

3.2.4 Data Handling

The demonstration software includes an interrupt driven data input routine and a section which will store data on disk.

The interrupt handler responds to an interrupt from the RSI CTS card by transferring the latest data frame from the CTS card to a large data buffer in RAM. It also keeps track of its place in the RAM buffers.

Two (2) data buffers of up to 64K length are allotted in the RAM on the Multifunction card in Screen #1 of the program. When one of these buffers is full, it is transferred to permanent media while the other is filling up. The size and addresses of the buffers you need will be determined by the type of permanent media you use and the rate at which the data is generated.

The data is stored in the data buffers in the form of 8 bit binary numbers. When this data is transferred to other media it is often convered into an ASCII character file. This is necessary to be consistent with floppy disk formats and to allow the use of word processing programs on the data. The data requires more room in the ASCII format. Some tape systems store the data directly as binary numbers.

3.3 Use of Commercial Software Packages

The following commercial software packages may be of help when used with this system.

3.3.1 Word Processor

Often it is necessary to edit, abbreviate, or pull out sections of the data files. These files are stored as ASCII characters and may be worked with most Word Processing Programs for the IBM PC. In the demonstration software a carriage return and line feed has been added at the end of each data frame to ease their reading with a word processor.

3.3.3 Graphics

Again, as the files are in ASCII characters, many commercial graphics packages will display the data files graphically. Graphics packages in Forth are also available for the IBM PC. These will facilitate development of data reduction and graphics display programs.

LMI makes a graphics extension for PC/FORTH

SPECIFIC APPLICATION SOFTWARE

File: CTSCUS

Introduction

The procedure for modifying the software for a specific application is detailed.

4.0 Specific Application Software

This section describes how to customize the general purpose software described in the last section for a specific application. Two (2) examples of custom software, for a single instrument and a dual instrument, are then given.

4.1 Hardware Set-up

Install the CTS card and any other peripherals you require in the IBM PC. Turn on the power, and see that a normal boot occurs.

Connect the cables to the IBM and external power supply but NOT to the flight instrument. Test the flight instrument cable connector pins to be sure that the power is on the correct pins. Turn off all power and connect the cables to the flight instrument.

You are now ready to turn the power back on and boot up the IBM PC.

4.2 Customizing the Software

Most of the customizations will be to adjust the program to your specific data requirements. You should, therefore, workup a Definition of Data Bits for your fight similar to the examples given in Appendix XVI.

Make a Copy of the CTS.SCR program using either the DOS COPY program or the FORTH SCOPY program with a new name for your instrument. Bring up FORTH and the EDITOR. Edit screen #0 to show the new program name and instrument.

Edit Screen #1 to set the segment address of BUF#1.DS AND BUF#2.DS. This sets up two 2K data buffers in high memory. The example values will be fine for most cases.

Edit the Forth word INIT.VAR on Screen #2 to set your values for DATA.RATE, in tens of Hertz; ENABLE.RATE, in Hz; and DATA.BITS. These should reflect your instrument's fight parameters and the Definition of the Data Bits developed earlier.

If the CTS card address or interrupt number have been changed, enter the changes also in Screen #2. This is only necessary if the card address conflicts with other cards in your system.

If you are using a separate card to read analog signals, Forth words to drive this card will be needed. These will probably be similar to the words on Screen #10. They might be placed on Screen #9, Screen #12, or the editor might be used to move screens to make room.

The display will now have to be modified to reflect the number and meaning or your data bits. Starting on Screen #27, edit the word .DATA, print data, to reflect the size and meaning of your data from your Definition of the Data Bits. You might also like to make changes such as expanding the display of the flag bits.

If you have a separate I/O card to read analog signals, you will need a word similar to .STATUS on Screen #28 to display the readings. This new word can then be added to DISPLAY.

Edit DISPLAY on Screen #29 if you want different information or the same information in a different arrangement to appear on the monitor during testing. You may also like to edit line four so that the name of the particular instrument appears on the screen during the test.

If you wish to store data on a device other than a disk under PCDOS control such as a large capacity tape, then the Data storage section on Screens #30 to #35 will require editing. Most large floppies and hard disks will work directly under the example program.

If you wish to use the Command Table feature, then enter your table in Screen #39. The elasped time in the first column is in seconds starting from the command to start the table. The '(tick) enters the time in memory. Your command is then entered. You may enter any defined FORTH word in the table but the stack will be empty. If you need to enter a word requiring parameters from the stack, then write a short word in Screen #38 which places the parameters on the stack and calls the word. Your new word can then be entered in the table. The word END.TABLE restarts the table when the last time entered has passed.

If your instrument, or data handling, require other special handling, you may modify any section of the example program to suit your needs. All modifications but those to the interrupt handler can be written in Forth.

You have now customized the CTS program for your application.

4.3 Running a Test

With the IBM PC up and booted, place the CTS software disk in Drive A and a formated disk for data in drive B. Enter the word FORTH followed by the name of your customized software. Load the program by entering 1 LOAD. When the ok prompt reappears on the monitor, enter RUN to start the test program.

The program will show the monitor display and ask for the name of a data file. Since this file will be on the B Disk the name should start with B:, as in B:DATAFILE.DAT . Your word processor may require a specific extension such as .DOC .

The program will then ask for a heading for the data file. The heading will appear at the top of the data file and may be up to 80 characters long. A carriage return will end the heading entry. The file name, experimenter, and test number are usually included.

The program will then show the standard display. The external power to the instrument should be turned on if it is not already on. At the bottom of the screen is an explanation of the keys which are active. The D will start the data clock and thereby start data reading. The S will start storage of data onto Disk as soon as a data buffer fills up. The number 0 to 7 will issue commands to the flight instrument.

Issue a few commands to insure the instrument is operating properly and that good data is coming through. Then turn on the data storage with a S and start the Command Table by entering a T. The incrementing of Elasped time will indicate that the command table is in charge. Occasionally, the disk drive will run and the Frames Stored number will increase. This indicates how much data has been saved.

Pressing any key will stop the command table and return control to the normal display. Entering a Q will then stop the Data Rate Clock, close the data file, and return operation to FORTH. External power to the flight instrument should then be cut off. The data file may then be reviewed and printed out with a word processor.

4.4 Single Instrument Software

The program CTR.SCR is set-up for a single spectrometer without scan platform. It is also used as the starting point for writing custom software for other instruments. The data format used is shown in Appendix XV or a single instrument.

4.5 Dual Instrument Software

The program CTRD.SCR is set-up for two (2) instruments with a scan platform. The data format is shown in Appendix XV under Dual Instrument with Platform.

The display has been expanded to show the second spectrograph data and the platform information.

The larger data word results in the buffers filling up quickly so the disk is called at shorter intervals.

CALIBRATION PROCEDURE

Introduction

The procedure for calibrating an instrument is detailed.

5.Ø Calibration Procedure

5.1 Set-up

5.2 Calibration Run

Appendix XIV

HARDWARE DETAILS
CALIBRATION AND TESTING SYSTEM
JOB 612 HUP2

J. T. Riley
04/07/88

File: CTSAPP

INTRODUCTION

Hardware details for the CTS card are given in detail including the I/O Map, Address Switch Settings, and Interrupt Jumper Settings.

1.0 I/O Map

This map assumes that the base address of 300 Hex is used.

Addr	Chip	Description
300		Clock 0 Data Rate
301		Clock 1 Enable Rate
302		Clock 2 counter for number of bits
303		Clock Mode address
304		Reset Data Rate Clock
305		Not used
306		Not used
307		Not used
308		Start Data Rate Clock
309		Not used
30A		Not used
30B		Not used
30C		Reset Enable
30D		Not used
30E		Not used
30F		Not used
310		Store Command / Read input
311		Not used
312		Not used
313		Not used
314		Read FIFO
315		Not used
316		Not used
317		Not used
318		Shift FIFO
319		Not used
31A		Not used
31B		Not used
31C		Clear FIFO

31D	Not used
31E	Not used
31F	Not used

2.0 Address Switch Settings

The bass address of the CTS card may be set using the DIP switch on the card. The setting are as follows.

Address Hex	Switch 1	2	3	4
200	On	On	On	On
220	On	On	On	Off
240	On	On	Off	On
260	On	On	Off	Off
280	On	Off	On	On
2A0	On	Off	On	Off
2C0	On	Off	Off	On
2E0	On	Off	Off	Off
300	Off	On	On	On
320	Off	On	On	Off
340	Off	On	Off	On
360	Off	On	Off	Off
380	Off	Off	On	On
3A0	Off	Off	On	On
3C0	Off	Off	Off	On

Switch 1 is away from the back connector.

3.0 Interrupt Jumper Settings

The interrupt may be set by setting the following on board jumpers. The location of these jumpers is shown in Figure 23, CTS Card Layout.

Shunt 3 — Enables Interrupt System
Alows testing without interrupts

Shunt 5	to	IRQ3	B25	Interrupt #3
	to	IRQ4	B24	Interrupt #4
	to	IRQ5	B23	Interrupt #5
	to	IRQ2	B4	Interrupt #2

The B connections are on the solder side of the card edge connector at the bottom of the card.

Appendix XV

DATA BIT IDENTIFICATION

J. T. Riley
03/07/88

Introduction

The data from the instrument is in the form of a large serial data word. The size of the data word is defined for likely configurations and the meaning of each bit given.

1.0 Word Length

The length of the data contributed by each microprocessor module must be divisible by eight. The exact length of the word and the meaning for each bit in the word is dependent on the needs of an individual flight. This is set in software by DATA (screen #34) and by the constant DATA# (screen #2).

The first instrument, a single spectrometer with a scan platform, used the following bits:

Spectrometer	
Count	13
Command	3
Wavelength	10
Status	6
Angle	13
Angle Status	2
Error	1
Total	48

The proposed configurations would then have the following word lengths.

One Spec without Scan	
Count	16
Wavelength	10
Status	6
Command	4

Error	4	
Total	40	5 Bytes
Two Spec. with Scan		
Master	40	
Scan	16	
Secondary	40	
Total	96	12 Bytes
Four Spec. no Scan		
Master	40	
Secondary	40	
#3	40	
#4	40	
Total	160	20 bytes

2.0 Data Timing

The microprocessor outputs data in response to a Data Timing (see Figure 24). The microprocessor expects these signals to be at TTL levels but they may be level shifted or inverted in the Spacecraft Interface module. Note that the Enable Signal must remain high during all the low to high transitions of the Data Clock. The Data Clock may be either gated or continuous. The microprocessor provides a new bit on each low to high transition of the clock and expects the spacecraft to read the bit just after each high to low.

The first bit of data will be available as soon as the Enable signal goes high, the Enable low to high transition must take place after a clock high to low. Similarly, the Enable signal must end after the last clock low to high and before the clock can go low to high again.

The Data Period, being the time between the start of data transfers, will be set by the spacecraft hardware and maybe any value between .01 and 1 seconds. If the data period is longer than the .01 second used on the first instrument, then additional Count bits will be needed (up to 16 bits) and the data word must then be extended. If the Data Period is longer than .1 second, the event counter may fill up and roll over on a bright source even with the full 16 bits.

The Data Clock rate is also set by the spacecraft and determines the time required to transfer the data. The total time require to transfer the data from all the instruments on a data channel should not exceed 70% of the Data Period or the system may not have time to execute commands properly. This rate may be any value between .5 KHz and 128 KHz. If the total data word is long, or the Data Period is short, the higher end of this range will be needed.

On multiple instrument configurations more than one data channel can be used to advantage. This will significantly improve data transfer rates. This feature would be implemented in hardware on the Spacecraft Interface card.

Software adjustment is necessary to set the Data Period for a mission (see DAT.EN.R screen 2), but the Data Clock rate does not affect software.

3.0 Bit Identification

The meaning of each bit of the Data can be set in software. The total length of Data for each microprocessor must be divisible by eight to comply with standard word length even if unchanging filler bits or extra error bits have to be added.

The number of Count Bits and the presents of a scan platform affect the total number of data bits directly.

The following example shows three (3) configurations. The first has a data bit pattern identical with the first engineering instrument, the second is a single instrument with no platform, and the third has two (2) instruments and a platform.

3.1 Spectrometer with Platform.

First Byte

1	Count, low word, bit 1, LSB
2	Count, low word, bit 2
3	Count, low word, bit 3
4	Count, low word, bit 4
5	Count, low word, bit 5
6	Count, low word, bit 6
7	Count, low word, bit 7
8	Count, low word, bit 8

Second Byte

9	Count, high word, bit 9
10	Count, high word, bit 10
11	Count, high word, bit 11
12	Count, high word, bit 12
13	Count, high word, bit 13, MSB
14	Command word, bit 1
15	Command word, bit 2
16	Command word, bit 3

Third Byte

17	Wavelength, low word, bit 1, LSB
18	Wavelength, low word, bit 2
19	Wavelength, low word, bit 3
20	Wavelength, low word, bit 4
21	Wavelength, low word, bit 5
22	Wavelength, low word, bit 6
23	Wavelength, low word, bit 7
24	Wavelength, low word, bit 8

Fourth Byte

25	Wavelength, high word, bit 9	
26	Wavelength, high word, bit 10, MSB	
27	Wavelength Fiducial.	1 = At fid
28	Cover Status,	1 = Closed

29	Shutter Status,	1 = Closed
30	Test Lamp Status,	1 = On
31	High Voltage Status,	1 = On
32	Solar Sensor Status	1 = Sun Present

Fifth byte

33	Angle, low word, bit 1, LSB
34	Angle, low word, bit 2
35	Angle, low word, bit 3
36	Angle, low word, bit 4
37	Angle, low word, bit 5
38	Angle, low word, bit 6
39	Angle, low word, bit 7
40	Angle, low word, bit 8

Sixth Byte

41	Angle, high word, bit 9	
42	Angle, high word, bit 10	
43	Angle, high word, bit 11	
44	Angle, high word, bit 12	
45	Angle, high word, bit 13, MSB	
46	Scan Angle Fiducial,	1 = At fid
47	Latch Status,	1 = Latched
48	Error Bit,	1 = Error.

This is the data bit definition used in the first instrument.

3.2 Spectrometer Alone

First Byte

1	Count, low word, bit 1, LSB
2	Count, low word, bit 2
3	Count, low word, bit 3
4	Count, low word, bit 4
5	Count, low word, bit 5
6	Count, low word, bit 6
7	Count, low word, bit 7
8	Count, low word, bit 8

Second Byte

9	Count, high word, bit 9
10	Count, high word, bit 10
11	Count, high word, bit 11
12	Count, high word, bit 12
13	Count, high word, bit 13,
14	Count, high word, bit 14
15	Count, high word, bit 15
16	Count, high word, bit 16, MSB

Third Byte

17	Wavelength, low word, bit 1, LSB

18	Wavelength, low word, bit 2
19	Wavelength, low word, bit 3
20	Wavelength, low word, bit 4
21	Wavelength, low word, bit 5
22	Wavelength, low word, bit 6
23	Wavelength, low word, bit 7
24	Wavelength, low word, bit 8

Fourth Byte

25	Wavelength, high word, bit 9	
26	Wavelength, high word, bit 10, MSB	
27	Wavelength Fiducial.	1 = At fid
28	Cover Status,	1 = Closed
29	Shutter Status,	1 = Closed
30	Solar Sensor Status	1 = Sun Present
31	High Voltage Status,	1 = On
32	Test Lamp Status,	1 = On

Fifth Byte

33	Command word, bit 1
34	Command word, bit 2
35	Command word, bit 3
36	Command word, bit 4
37	Error Bit
38	Error Bit
39	Error Bit
40	Error Bit

Compared to the first instrument, the count has been extended to 16 bits to allow longer intergration times. The Command Word has been moved to the last byte and extended by one bit so that it can show the exact command number instead of the command number minus one.

3.3 Dual Spectrometers

Dual spectrometers with a scanning platform controlled by the master unit.

First Byte

1	Count, low word, bit 1, LSB
2	Count, low word, bit 2
3	Count, low word, bit 3
4	Count, low word, bit 4
5	Count, low word, bit 5
6	Count, low word, bit 6
7	Count, low word, bit 7
8	Count, low word, bit 8

Second Byte

9	Count, high word, bit 9
10	Count, high word, bit 10
11	Count, high word, bit 11

12	Count, high word, bit 12	
13	Count, high word, bit 13,	
14	Count, high word, bit 14,	
15	Count, high word, bit 15,	
16	Count, high word, bit 16, MSB	
Third Byte		
17	Wavelength, low word, bit 1, LSB	
18	Wavelength, low word, bit 2	
19	Wavelength, low word, bit 3	
20	Wavelength, low word, bit 4	
21	Wavelength, low word, bit 5	
22	Wavelength, low word, bit 6	
23	Wavelength, low word, bit 7	
24	Wavelength, low word, bit 8	
Fourth Byte		
25	Wavelength, high word, bit 9	
26	Wavelength, high word, bit 10, MSB	
27	Wavelength Fiducial.	1 = At fid
28	Cover Status,	1 = Closed
29	Shutter Status,	1 = Closed
30	Solar Sensor Status	1 = Sun Present
31	High Voltage Status,	1 = On
32	Test Lamp Status,	1 = On
Fifth Byte		
33	Command word, bit 1	
34	Command word, bit 2	
35	Command word, bit 3	
36	Command word, bit 4	
37	Error Bit	
38	Error Bit	
39	Error Bit	
40	Error Bit	
Sixth byte		
33	Angle, low word, bit 1, LSB	
34	Angle, low word, bit 2	
35	Angle, low word, bit 3	
36	Angle, low word, bit 4	
37	Angle, low word, bit 5	
38	Angle, low word, bit 6	
39	Angle, low word, bit 7	
40	Angle, low word, bit 8	
Seventh Byte		
41	Angle, high word, bit 9	
42	Angle, high word, bit 10	
43	Angle, high word, bit 11	
44	Angle, high word, bit 12	
45	Angle, high word, bit 13, MSB	
46	Scan Angle Fiducial,	1 = At fid

47	Latch Status,	1 = Latched
48	Error Bit,	1 = Error.

Eight Byte — Start of Secondary Instrument

1	Count, low word, bit 1, LSB
2	Count, low word, bit 2
3	Count, low word, bit 3
4	Count, low word, bit 4
5	Count, low word, bit 5
6	Count, low word, bit 6
7	Count, low word, bit 7
8	Count, low word, bit 8

Ninth Byte

9	Count, high word, bit 9
10	Count, high word, bit 10
11	Count, high word, bit 11
12	Count, high word, bit 12
13	Count, high word, bit 13
14	Count, high word, bit 14
15	Count, high word, bit 15
16	Count, high word, bit 16, MSB

Tenth Byte

17	Wavelength, low word, bit 1, LSB
18	Wavelength, low word, bit 2
19	Wavelength, low word, bit 3
20	Wavelength, low word, bit 4
21	Wavelength, low word, bit 5
22	Wavelength, low word, bit 6
23	Wavelength, low word, bit 7
24	Wavelength, low word, bit 8

Eleventh Byte

25	Wavelength, high word, bit 9	
26	Wavelength, high word, bit 10, MSB	
27	Wavelength Fiducial.	1 = At fid
28	Cover Status,	1 = Closed
29	Shutter Status,	1 = Closed
30	Solar Sensor Status	1 = Sun Present
31	High Voltage Status,	1 = On
32	Test Lamp Status,	1 = On

Twelveth Byte

33	Command word, bit 1
34	Command word, bit 2
35	Command word, bit 3
36	Command word, bit 4
37	Error Bit
38	Error Bit
39	Error Bit
40	Error Bit

The count is extended to 16 bits to allow longer intergration times. The Command Word is moved to the last byte and extended by one bit so that it can show the exact command number instead of the command number minus one.

The data bit identifications for the two (2) instruments are identical, but two bytes of platform data have been added to the master instrument.

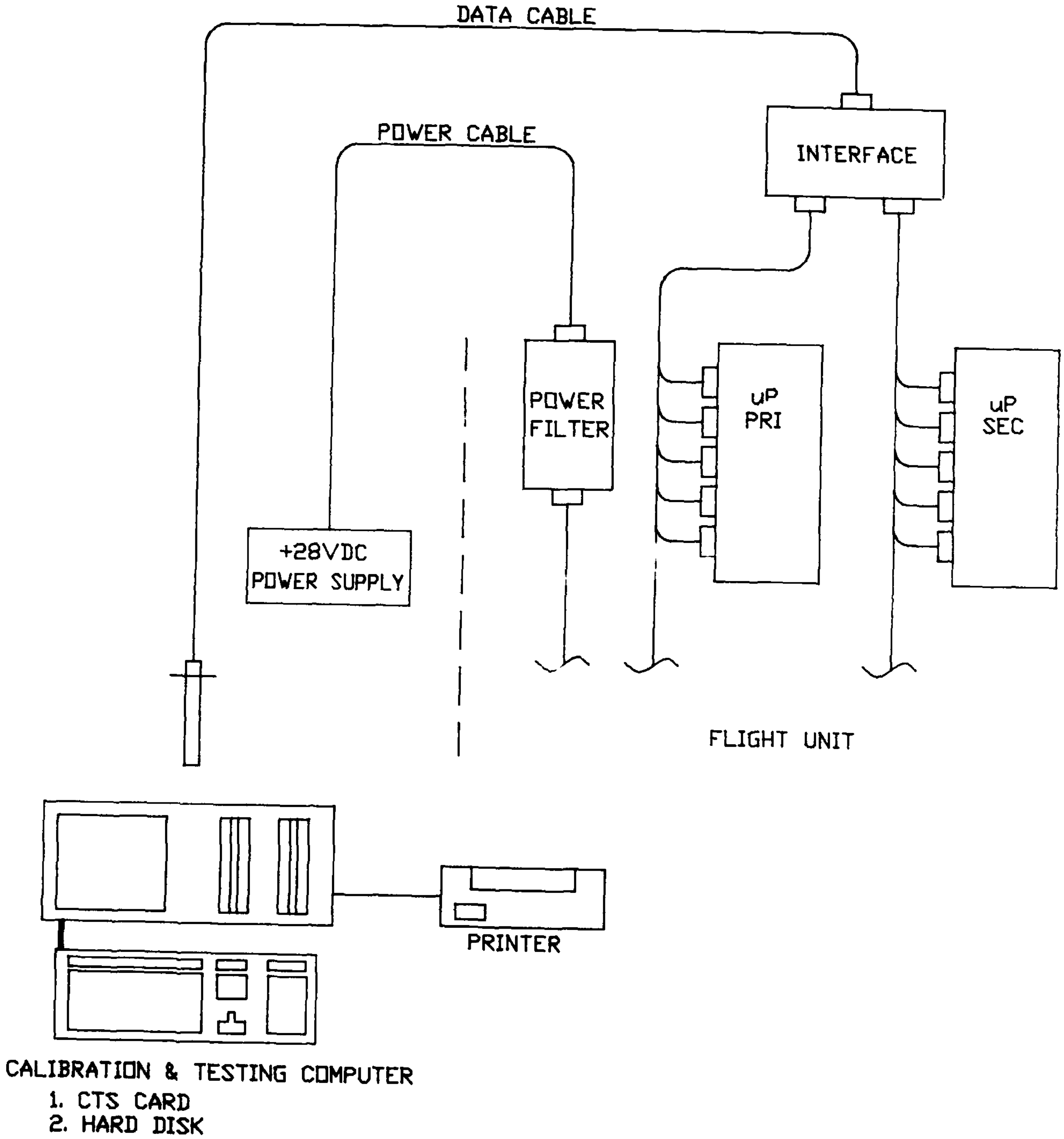


FIG 22 CALIBRATION &TESTING SYSTEM (CTS)

RILEY
1-11-89

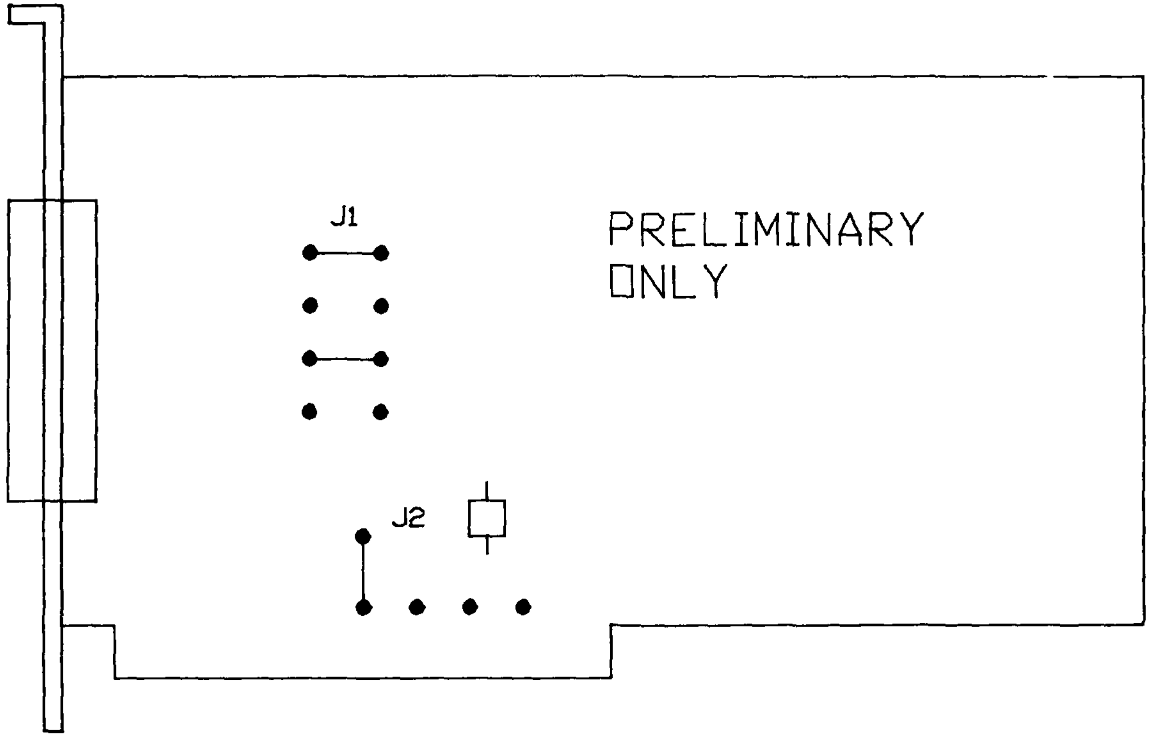
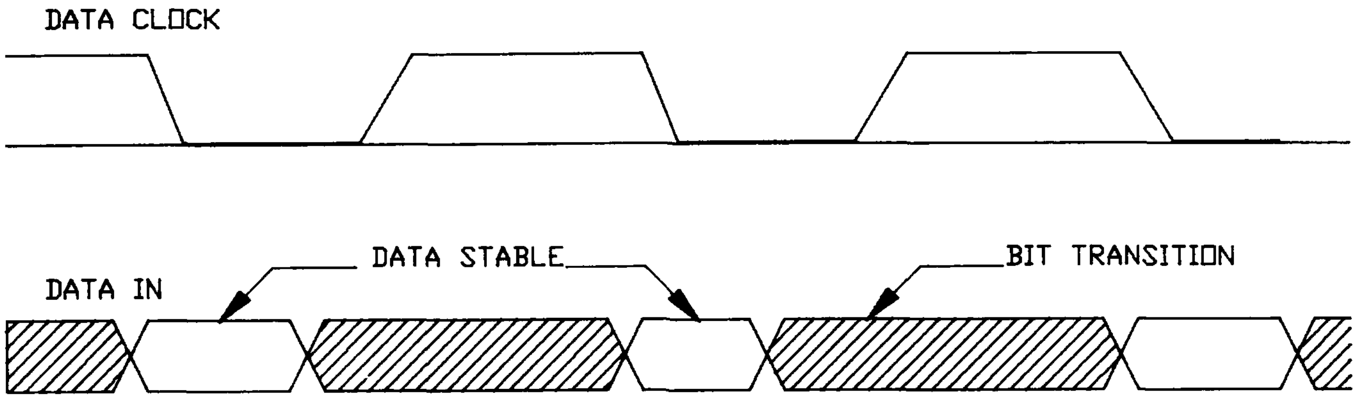


FIG 23 RSI CTS CARD LAYOUT

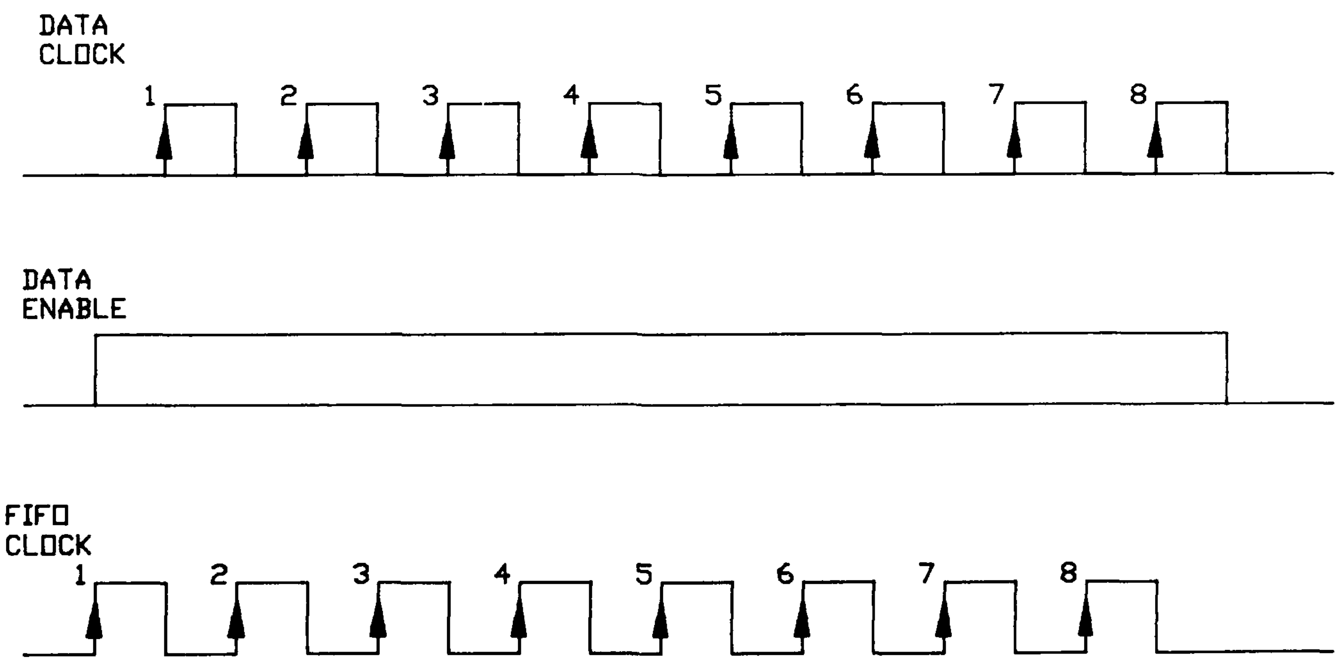
FIG 24 DATA TIMING

TIME FOR DATA STABLITY



CLOCK RELATIONS

SYSTEM SET FOR 8 DATA BITS
1 BITS READ



```
Screen # 0
( CALIBRATION & TESTING SYSTEM  CTS              JTR 10:57 04/08/88 )
( Last change:   Screen  009                     JTR 11:17 04/08/88 )

APPENDIX  XVI
                   SINGLE INSTRUMENT SOFTWARE
                        no platform
                        VERSION 1.1

Copyright (c) 1985
Research Support Instruments, Inc.
10610 Beaver Dam Road
Cockeysville, Maryland  21030

Written by:   Tom Riley

FILE:  CTS.SCR


Screen # 1
( Variables                                     JTR 10:57 04/08/88 )
: MARK ;                ( Marks start of application words )
VARIABLE  DATA.RATE   ( Data clock rate in DecaHz, 10's )
VARIABLE  ENABLE.RATE   ( Enable Rate in Hz )
VARIABLE  DATA.BITS   ( Number of data bits )
VARIABLE  CUR.COM    0 CUR.COM !         ( Current Command)
VARIABLE  DATA.BYTES  ( Number of data Bytes )
     HEX
VARIABLE  DATA.POINT          ( Data Pointer )
4000 CONSTANT BUF#1.DS        ( Buffer #1 DS Register )
4800 CONSTANT BUF#2.DS        ( Buffer #2 DS Register )
VARIABLE  BUF.LEN             ( Buffer Length )
VARIABLE  NEX.BUF             ( Next Buffer DS )
VARIABLE  PRES.BUF            ( Present Buffer DS )
VARIABLE  NEX.BUF.STOR        ( Next Buffer to Disk DS )
     DECIMAL -->


Screen # 2
( INIT.VAR, CTS.ADDR                            JTR 10:58 04/08/88 )

: INIT.VAR    (  --  . Initialize application variables )
        DECIMAL                    ( Change initial values here )
        1000    DATA.RATE !          ( DecaHz, 10,000 Hz )
        10      ENABLE.RATE !        ( Hz  )
        40      DATA.BITS !          ( bits )
         2  1024 * DATA.BITS @ - 1 -   BUF.LEN !    ( Bytes )
        DATA.BITS @ 15 + 16 /  2 * DATA.BYTES !   ;
                 ( divisiable by 8, even number )
HEX
300   CONSTANT   CTS.ADDR     ( Base address of CTS card )
3     CONSTANT   INTERRUPT#       ( Interrupt Number )
DECIMAL -->
```

```
Screen # 3
( Store Data Rate Parameter                         JTR 09:40 02/25/88 )
ASM86  HEX  ( Load assembler )

CODE  DR!   ( N -- , Store Data Rate Parameter on card )
       DX, # CTS.ADDR 3 + MOV          ( Clock Mode Port Address)
       AL, # 36           MOV  ( Set Mode, Clk 0 Sq Wav)
       DX,  AL            OUT
       DX, # CTS.ADDR       MOV          ( Parameter Port )
       AX       POP                    ( get n from stack)
       DX,    AL          OUT          ( LSB )
       AL,    AH          MOV         ( Next BYTE)
       DX,    AL          OUT         ( MSB )
       NEXT, END-CODE      -->



Screen # 4
( Store Enable Rate Parameter                       JTR 09:40 02/25/88 )

CODE  ER!   ( N -- , Store Enable Rate Parameter on card )
       DX, # CTS.ADDR 3 + MOV          ( Clock Mode Port Address)
       AL, # 74           MOV  ( Set Mode, CLK 1 RATE GEN )
       DX, AL             OUT
       DX, # CTS.ADDR 1+  MOV          ( Parameter Port )
       AX       POP                    ( get n from stack)
       DX,    AL          OUT          ( LSB )
       AL, AH  MOV   DX, AL OUT          ( MSB)
       NEXT, END-CODE             -->



Screen # 5
( Store Bit Count Parameter                         JTR 09:40 02/25/88 )

CODE  BN!   ( N -- , Store Number of Bits on card )
       DX, # CTS.ADDR 3 + MOV          ( Clock Mode Port Address)
       AL, # B4           MOV  ( Set Mode, CLK 2 RATE GEN )
       DX, AL             OUT
       DX, # CTS.ADDR 2 + MOV          ( Parameter Port )
       AX                 POP          ( get n from stack)
       DX.    AL          OUT          ( LSB )
       AL,    AH          MOV         ( Next BYTE)
       DX.    AL          OUT          ( MSB )
       NEXT, END-CODE

  -->


Laboratory Microsystems PC/FORTH 3.00      13:44   10/30/89   a:cts.scr
```

```
Screen # 6
( Start Data Rate, Reset Data Rate             JTR 11:17 04/08/88 )

CODE  START.DR   (  -- , Start Data Rate)
        DX, # CTS.ADDR 8 +  MOV    ( Port Address )
        AL, # 00            MOV    ( Any number )
        DX,   AL            OUT    ( write to address)
        NEXT, END-CODE

CODE  STOP.DR   (  -- , Reset Data Rate)
        DX, # CTS.ADDR 4 + MOV    ( Port Address )
        AL, # 00            MOV    ( Any number )
        DX,   AL            OUT    ( write to address)
        NEXT, END-CODE


              -->


Screen # 7
( Reset Enable, Send Command                   JTR 11:17 04/08/88 )

CODE  RESET.E   (  -- , Reset Enable)
        DX, # CTS.ADDR C + MOV    ( Port Address )
        AL, # 00            MOV    ( Any number )
        DX,   AL            OUT    ( write to address)
        NEXT, END-CODE

CODE  COMMAND!   ( N  -- , Issue Command  )
        AX      POP                    ( get n from stack)
        DX, # CTS.ADDR 10 +   MOV    ( Port Address )
        DX,   AL            OUT    ( write to address)
        NEXT, END-CODE

      -->


Screen # 8
( Read FIFO, Shift FIFO                        JTR 11:17 04/08/88 )

CODE  @FIFO   (  -- N , Read FIFO)
        DX, #  CTS.ADDR 14 + MOV    ( Port Address )
        AL,  DX              IN     ( Read 1 Byte )
        AH, # 0              MOV
        AX                   PUSH
        NEXT, END-CODE

CODE  SHIFT.FIFO   (  -- , Parallel Shift FIFO for next byte )
        DX. #  CTS.ADDR 18 + MOV    ( Port Address )
        AL, # 00            MOV    ( Any number )
        DX,   AL            OUT    ( write to address)
        NEXT, END-CODE

                          -->
```

```
Screen # 9
( Clear FIFO                                      JTR 11:17 04/08/88 )

CODE  CLEAR.FIFO    (  -- , Clear FIFO completely )
        DX, # CTS.ADDR 1C + MOV     ( Port Address )
        AL, # 00            MOV     ( Any number )
        DX,   AL            OUT     ( write to address)
        NEXT. END-CODE


DECIMAL
             -->
```

```
Screen # 10
( SET.DATA.RATE, SET.ENABLE.RATE                  JTR 12:30 10/30/85 )

: SET.DATA.RATE   ( N -- , Set Data Rate )
        2385 SWAP 100 SWAP  */  ( convert to parameter )
        DR! ;

: SET.ENABLE.RATE   ( N --   , Set Enable Rate )
        DATA.RATE @ 10 4 */
        SWAP /
        ER! ;

: SET.DATA.BITS   ( N  --   , Set number of data bits )
        DUP 256 > IF DROP 256 THEN     ( less than 256)
        BN! ;               ( store )
-->
```

```
Screen # 11
( INIT.CTS                                        JTR 12:32 10/30/85 )

: INIT.CTS    (    --   , Initiate CTS card )
        INIT.VAR
        STOP.DR
        RESET.E
        0 COMMAND!
        CLEAR.FIFO
        DATA.RATE @  SET.DATA.RATE
        ENABLE.RATE @  SET.ENABLE.RATE
        DATA.BITS @  SET.DATA.BITS
         ;


  -->
```

```
Screen # 12
( WAIT, MSEC                                       JTR 10:58 04/08/88 )

: WAIT   ( N --   , Short delay )
        0 DO LOOP ;

: MSEC    ( N -- , delay in milliseconds )
        0 DO  36 WAIT  LOOP ;

 -->
```

```
Screen # 13
( SEND.COM                                         JTR 13:20 02/24/88 )

: SEND.COM   ( N --   , Send CTS command as Level )
        DUP 8 > IF DROP 0 THEN     ( Less than 8)
        DUP 1 < IF DROP 0 THEN     ( Greater than 1 )
        DUP  CUR.COM !             ( Current command )
        2 SWAP BEGIN DUP WHILE
             1- SWAP 2* SWAP REPEAT        ( Shift to bit )
        DROP 2/ 2/    COMMAND! ;
: PULSE.COM  ( N --   , Send CTS command as Pulse 0 )
        DUP CUR.COM !             ( Update Current Command )
        DUP SEND.COM              ( Set Bit)
        100 MSEC                   ( wait )
    (   0  SEND.COM   )           ( 0 Bits )
        CUR.COM !                 ( Store last command )
        ; -->
```

```
Screen # 14
( FDOS.EX                                          JTR 11:37 12/26/88 )
HEX

CODE FDOS.EX   ( AX, DS, DX -- AX, BX , Extended FDOS )
        DX POP DS POP AX POP    ( Load Parameters )
        SI PUSH BP PUSH           ( Conserve pointers )
        21 INT                    ( Call DOS Interrupt )
        BP POP SI POP             ( Restore Pointers )
        AX PUSH BX PUSH           ( Return results )
        NEXT, END-CODE



  -->
```

```
Screen # 15
( Set-up Interrupt                              JTR 11:05 04/08/88 )

CODE @CS     (    --   N , Fetch CS register )
        CS PUSH    NEXT, END-CODE

CODE @DS     (    --   N , Fetch DS register )
        DS PUSH    NEXT, END-CODE

CODE SOFT.INT  (     -- , Call Interrupt from Software )
        13      INT
        NEXT, END-CODE

-->
```

```
Screen # 16
( @IMR, !IMR, RESET. MASK of 8259A          JTR 10:58 04/08/88 )
HEX
CODE @IMR  (  -- N, Read Interrupt Mask Register of 8259A )
        # 21 , AL  IN
        AH, # 0 MOV
        AX PUSH
        NEXT, END-CODE
CODE !IMR  ( N -- , Store Interrupt Mask Register )
        AX    POP
        # 21 , AL  OUT  STI  ( Enable interrupt )
        NEXT, END-CODE



-->
```

```
Screen # 17
( RESET.MASK, MASK  8259A                       JTR 09:41 02/25/88 )

: RESET.MASK  (  -- , Reset 8259 Interrupt Mask )
        @IMR                            ( Read Mask )
        1 INTERRUPT# SHIFT   NOT        ( 0 in correct bit )
        AND !IMR ;                      ( Store Mask )


: MASK  (  -- , Mask 8259 Interrupt Mask )
        @IMR                            ( Read Mask )
        1 INTERRUPT# SHIFT              ( 1 in correct bit )
        OR !IMR ;                       ( Store Mask )

        -->
```

```
Screen # 18
( Interrupt Service  1                              JTR 15:54 03/20/86 )
HEX
CODE INTER.SERV ( -- , Assembly Interrupt Service Routine )
        PUSHF AX PUSH  BX PUSH  CX PUSH  DX PUSH  ( Save  )
        BP PUSH SI PUSH DS PUSH  DS PUSH ( System Configure )
        AX,  CS  MOV   DS,  AX  MOV ( CS to DS )
        AX,   DATA.BYTES MOV  CX,  AX MOV ( Count Data Bytes )

1$:     AX, DATA.POINT  MOV  BX, AX  MOV  ( Data Pointer )
        AX, PRES.BUF  MOV   DS,  AX  MOV  ( Data Segment )
        DX, # CTS.ADDR 14 + MOV    ( READ FIFO )
        AL,  DX               IN    ( Read FIFO )
        [BX],  AL     MOV    ( Store Byte )
        DX, # CTS.ADDR 18 +  MOV   ( Shift FIFO )
        DX,  AL               OUT
      -->


Screen # 19
( Interrupt Service Cont.  2                        JTR 11:58 02/12/86 )
        AX,  CS  MOV   DS,  AX  MOV ( CS to DS )
        WORD  DATA.POINT  INC       ( Increment Data Pointer )
        CX      DEC                 ( Reduce count )
        AX,  CX  MOV    AX,   AX  AND  ( Set flag)
        1$              JNZ

        DX,  # CTS.ADDR 1C +  MOV   ( CLEAR FIFO )
        DX,  AL                OUT  ( Clear FIFO )

        AX,  CS  MOV   DS,  AX  MOV ( CS to DS )
        AX,  BUF.LEN    MOV         ( End of Buffer )
        BX,  DATA.POINT MOV
        AX, BX          CMP
        2$      JNL
 -->


Screen # 20
( Interrupt Service Cont. 3                         JTR 15:55 03/20/86 )
        AX,    PRES.BUF MOV         ( Swap DS of Present )
        BX,    NEX.BUF  MOV         ( and Next Buffer )
        PRES.BUF , BX   MOV
        NEX.BUF ,  AX   MOV
        AX,  # 00       MOV      ( Zero Pointer )
        DATA.POINT , AX MOV
2$:     DX,  # CTS.ADDR  C + MOV  ( Reset Card Interrupt )
        DX,  AL        OUT        ( Any number out )
        DX, # 20    MOV  AL, # 20  MOV    ( Reset 8259A )
        DX,  AL   OUT

        DS POP  CS POP  SI POP BP POP
        DX POP  CX POP  BX POP  AX POP POPF   ( Restore )
        IRET  END-CODE            ( Return from Interrupt )
    DECIMAL  -->
```

```
Screen # 21
(                                                    16:20 02/12/86 )















  --)




Screen # 22
( SET.BUFFERS                                    JTR 10:59 04/08/88 )
HEX
: SET.INTER.VEC ( --   , Set Interrupt Vector )
        25 100 * INTERRUPT# 8 + +
        @CS ['] INTER.SERV 2 +        ( Sector and Code Addr)
       FDOS.EX  DROP DUP
           ;



: SET.BUFFERS  ( -- , Set-up Data Buffers )
        BUF#1.DS   PRES.BUF !
        BUF#2.DS   NEX.BUF !
        BUF#1.DS   NEX.BUF.STOR !
        0 DATA.POINT ! ;
     --)

Screen # 23
( Interrupt test                                 JTR 10:59 04/08/88 )
HEX
: INIT.INTER ( -- , Initiate Interrupt Routine )
     SET.INTER.VEC                 ( Set-up Interrupt Vec )
     SET.BUFFERS                   ( Set-up Data Buffers )
     RESET.MASK                    ( Enable interrupt Mask )
     ;
DECIMAL

: TEST  ( N --   , INTERRUPT TEST )
        0 DO
                DATA.POINT @ U. PRES.BUF @ U. CR
        LOOP ;


    --)

Laboratory Microsystems PC/FORTH 3.00      13:47  10/30/89   a:cts.scr
```

```
Screen # 24
( Display and Command Table Variables       JTR 11:02 04/08/88 )

VARIABLE  ?DATA 0 ?DATA ! ( Data taking Flag 0 - Taking Data )
VARIABLE ?STORING  0 ?STORING ! ( Storage Flag 0 - Storing )
VARIABLE STOR.POINT      ( Data storage memory pointer )
VARIABLE  FRAM.STOR  0 FRAM.STOR ! ( Data Frames Stored )
VARIABLE   ELAPS.TIME      ( Elapsed time for table, sec )
VARIABLE   CUR.ENT         ( Current Entry from table )
VARIABLE   START.TIME      ( Start time for table, sec )

: TOG.DATA  (   -- , Toggle ?DATA )
        ?DATA @  NOT DUP ?DATA !      ( Set Flag )
        IF START.DR ELSE STOP.DR THEN ; ( Start/stop Data Rate)

: TOG.STOR ( --   ,  Toggle ?STORING )
        ?STORING @ NOT ?STORING ! ;  -->


Screen # 25
( @DATA.                                     JTR 09:41 02/25/88 )

CODE @DATA ( ADDR Seg -- N, Fetch Data from data buffer )
        DS       POP                ( Get Segment )
        BX       POP                ( Get Data Address )
        AX,      [BX]    MOV       ( Get 2 Bytes )
        BX,      AX      MOV       ( Save )
        AX,      CS      MOV       ( Restore DS )
        DS,      AX      MOV
        BX               PUSH      ( Place on Stack )
        NEXT,  END-CODE

: ON/OFF   ( F --   , Print ON  or OFF )
        IF ." ON   " ELSE ." OFF " THEN ;
-->


Screen # 26
( @LAT.DATA                                  JTR 14:53 02/2_/88 )

: @LAT.DATA (   -- Nn ... N1, Fetch Latest Data )
        DATA.BYTES @
        0  DO
                DATA.POINT @  I - 1 -    PRES.BUF @
                @DATA
           2 +LOOP ;

         -->
```

```
Screen # 27
( .DATA                                          JTR 12:47 02/25/88 )

: .DATA    ( --    , Display Latest Data )
        @LAT.DATA
        ." Count           : " 8 U.R  CR
        ." Wavelength      : " DUP 1023 AND 8 U.R  CR
        ." Flags           : " 256 / 255 AND 8 BINARY U.R
                                    DECIMAL CR
        ." Command         : " 256 / 15 AND 8 U.R CR CR
             ;          --)
```

```
Screen # 28
( .DATE/TIME                                     JTR 11:02 04/08/88 )

: .DATE/TIME ( -- , Print Date and time )
        @DATE  256 /MOD SWAP
             ." Day " . ." . Month " .
             ." , Year " . 8 SPACES
        @TIME SWAP 256 /MOD  .
        ." : " . 256 /MOD ." : " . DROP
        9 SPACES  CR ;

: .STATUS    (  --   , Status Display )
        CUR.COM @  ." Current Command - " . CR
        ?DATA @    ." Data Taking      - " ON/OFF 12 SPACES
        ?STORING @ ." Storing Data     - " ON/OFF  CR
        ELAPS.TIME @ ." Elapsed Time    - " 8 U.R 8 SPACES
        FRAM.STOR @ ." Frames Stored   - " U. 4 SPACES CR ; --)
```

```
Screen # 29
( DISPLAY , COM.LINE                             JTR 12:49 02/25/88 )

: DISPLAY   ( --   , Standard display )
        0 0 GOTOXY
        10 SPACES ." RSI    CALIBRATION AND TESTING " CR CR
        .DATE/TIME CR
        .DATA      CR
        .STATUS    CR
 ." Active Keys - " CR
 ."    0 - 7  Send Command " CR
 ."    D -    Toggle Data Taking      T - Run Command Table" CR
 ."    S -    Toggle Data Storage     Q - Quit Program " CR ;

: COM.LINE (  --    , Move cursor to clear command line )
     0 23 GOTOXY 60 SPACES 0 23 GOTOXY ;
--)
```

```
Screen # 30
( File Storage Support                          JTR 13:43 02/12/92 )
DOSINT     ( Load PCDOS vocabulary )

HANDLE FILE1
VARIABLE WBUF  100 ALLOT  ( Write Buffer )
VARIABLE WLEN  0 WLEN !     ( Length of WBUF text string )
: WRITE.LINE  (  --  , Write line from WBUF to disk )
        FILE1 WLEN @ DUP >R  WBUF  WRITE
        IF COM.LINE ." Write error " CR QUIT THEN
        R> <>
        IF CR ." Media Full " CR QUIT THEN
        0 WLEN ! ;
: ADD.SP  ( -- , Add space to WBUF )
     32 WBUF WLEN @ +  C!     ( ASCII Space 32 )
      1  WLEN  +! ;
                                                 -->


Screen # 31
( Storage support                               JTR 09:41 02/25/88 )

: ADD.NUM    ( N --   , Adds number to WBUF )
        0 <# #S #>
        0 DO  DUP C@ WBUF WLEN @ + C!
           1  WLEN  +! 1 + LOOP DROP ADD.SP ;
: ADD.CR  ( -- , Add CR and LF to WBUF )
        13 WBUF WLEN @ + DUP ROT ROT C! ( CR )
        1 + 10 SWAP C!                  ( LF )
        2 WLEN  +! ;              ( count )


: OPEN.IT         FILE1 OPEN-FILE
                  IF COM.LINE ." Can't open file" CR QUIT THEN ;

                                                      -->


Screen # 32
( Storage support                               JTR 13:56 02/12/92 )
: GET-FILENAME  ( -- , Enter File name )
                COM.LINE
                ." Enter Path and Filename:  "
                FILE1 INPUT-FILENAME  CR
                FILE1 MAKE-FILE DROP OPEN.IT ;

: CLOSE.IT      FILE1 CLOSE-FILE DROP ;

: ADD.DATE  (  --  , Add Date and Time to WBUF )
        ADD.CR @DATE SWAP ADD.NUM
        256 /MOD  ADD.NUM ADD.NUM
        @TIME SWAP 256 /MOD ADD.NUM ADD.NUM
        256 / ADD.NUM  ADD.CR  ADD.CR
        WRITE.LINE ;                              -->
```

```
Screen # 33
( FILE.HEAD                                        JTR 13:15 02/25/86 )

: FIND.LEN (   --    , Find Length of WBUF )
        0 WLEN ! WBUF 0 BEGIN
               2DUP  + C@ WHILE
               1 WLEN +! 1 + REPEAT DROP DROP ;

: FILE.HEAD    (  --  , Get file Heading   )
        COM.LINE  ." Enter File Heading : "
        WBUF 64 DUP WLEN ! EXPECT      ( Get Text String )
        FIND.LEN ADD.CR ADD.CR         ( Number of Char )
        WRITE.LINE ;


-->



Screen # 34
( STORE.BUF                                        JTR 11:01 04/08/88 )

: STORE.BUF    ( --   , Store buffer on disk )
        0 STOR.POINT !    ADD.DATE
        BUF.LEN @ 0 DO                 ( Entire buffer )
            FRAM.STOR @ ADD.NUM
            1 FRAM.STOR +!             ( Frame count )
            DATA.BYTES @ 0 DO          ( One data Frame )
              NEX.BUF.STOR @ STOR.POINT @ @L  ( Get Datum )
              ADD.NUM                 ( Add to WBUF )
              2 STOR.POINT  +!
              2 +LOOP ADD.CR              ( Add CR )
            WRITE.LINE
        DATA.BYTES @ +LOOP             ( Next Data Frame )
        PRES.BUF @   NEX.BUF.STOR ! ;
-->


Screen # 35
( STORE.BUFF                                       JTR 09:43 02/25/88 )

: ?STORE.BUFF (  --   , Store Buffer to Disk )
        ?STORING @                          ( Storing ? )
           NEX.BUF @  NEX.BUF.STOR @ = AND   ( Buf Full ? )
               IF    STORE.BUF             ( Store buffer )
        THEN ;




: STOP    (    --   , Stop the program )
        CLOSE.IT  STOP.DR MASK COM.LINE ;

                                              -->
```

```
Screen # 36
( TABLE Support                                           JTR 11:02 04/08/88 )

: ENTER   (  N ADDR --   , Make entry into table )
         SWAP , , ;    ( Move address and time into memory )

: @SEC   (  -- N, Get number of seconds today )
        @TIME 256 /MOD SWAP DROP SWAP    ( Get seconds )
           256 /MOD 6 - 60 * + 60 * + ; ( Hours and minutes )

: !START.TIME   (  --   , Get start time )
        @SEC  START.TIME ! ;

: @ELAPS.TIME (  -- N , Get elasped time )
        @SEC START.TIME @ - DUP ELAPS.TIME ! ;

-->
```

```
Screen # 37
( SERVICE TABLE                                           JTR 11:02 04/08/88 )

: RESTART.TABLE ( -- , Restart Table )
        !START.TIME  @ELAPS.TIME DROP 0  CUR.ENT ! ;
: END.TABLE   (  --   , End the Table )
        0  ['] RESTART.TABLE  ENTER ;    ( Last entry )

                             -->
```

```
Screen # 38
( Table Commands                                          JTR 09:43 02/25/88 )


: STORE ( --   ,  Start Data Storing )
        -1 ?STORING ! ;

: COM.#1 ( -- ,   Issue Command # 1 )
        1 PULSE.COM ;
: COM.#0 ( -- ,   Issue Command # 0 )
        0 PULSE.COM ;



-->
```

```
Screen # 39
( COMMAND.TABLE                                          JTR 09:43 02/25/88 )

VARIABLE  COMMAND.TABLE     -2 ALLOT    ( Memory Heading )
( Time    '   Command        ENTER       )
  10      '   COM.#1         ENTER
  20      '   COM.#0         ENTER
  30      '   COM.#1         ENTER
  40      '   COM.#0         ENTER


END.TABLE -->
```

```
Screen # 40
( Service Command table                                  JTR 11:03 04/08/88 )

: TABLE.EXEC   ( N-- , Execute from Table )
        COMMAND.TABLE + 2 + @       ( Calculate address )
             EXECUTE                ( Execute the command )
        CUR.ENT @ 4 + CUR.ENT ! ; ( Iterate Current Entry )
: TIME.CHECK   ( -- F or N , T . Compare Time )
        COMMAND.TABLE CUR.ENT @ + @  @ELAPS.TIME   <=
        IF   CUR.ENT @ -1  ELSE 0 THEN ;
: COMMANDS ( --   ,   Run Command Table )
      RESTART.TABLE BEGIN  ?TERMINAL NOT WHILE
          TIME.CHECK  IF TABLE.EXEC   ( time for next com ? )
                   ELSE DISPLAY  ( Refresh display )
                        ?STORE.BUFF
          THEN REPEAT
          KEY DROP 0 ELAPS.TIME ! ; ( Any key to stop ) -->
```

```
Screen # 41
( KEY Functions                                          JTR 13:58 02/12/92 )

: SERV.KEY     ( -- , Service Keyboard )
        ?TERMINAL   IF KEY           ( Key pressed ? )
           DUP DUP ASCII 0 >= SWAP ASCII 7 <= AND  ( 0 - 7 ? )
                IF DUP ASCII 0 - PULSE.COM THEN  ( Send Com )
           DUP ASCII D  = IF                   ( D ? )
                TOG.DATA  THEN                 ( Set flag )
           DUP ASCII S  = IF                   ( S )
                TOG.STOR  THEN                 ( Set flag )
           DUP ASCII T  = IF                   ( T )
                COMMANDS  THEN                 ( Run Table )
           DUP ASCII Q = IF
              STOP QUIT   THEN                 ( Quit )
           DROP THEN ;
-->
```

Screen # 42

```
( RUN                                          JTR 11:04 04/08/88 )

: RUN    (  --   , Run test   )
        INIT.CTS                          ( Initiate system )
        INIT.INTER
        CLEARSCREEN  DISPLAY              ( Set-up Monitor )
        GET-FILENAME FILE.HEAD            ( Set-up Disk File )
        BEGIN  1 WHILE                    ( Infinite loop )
           DISPLAY
           SERV.KEY
           ?STORE.BUFF
        REPEAT ;                  ( Exit on key Q )
```

Screen # 43

Screen # 44

```
  0 ( CALIBRATION & TESTING SYSTEM  CTS          JTR 10:57 04/08/88 )
  1 ( Variables                                  JTR 10:57 04/08/88 )
  2 ( INIT.VAR, CTS.ADDR                         JTR 10:58 04/08/88 )
  3 ( Store Data Rate Parameter                  JTR 09:40 02/25/88 )
  4 ( Store Enable Rate Parameter                JTR 09:40 02/25/88 )
  5 ( Store Bit Count Parameter                  JTR 09:40 02/25/88 )
  6 ( Start Data Rate, Reset Data  Rate          JTR 11:17 04/08/88 )
  7 ( Reset Enable, Send Command                 JTR 11:17 04/08/88 )
  8 ( Read FIFO, Shift FIFO                      JTR 11:17 04/08/88 )
  9 ( Clear FIFO                                 JTR 11:17 04/08/88 )
 10 ( SET.DATA.RATE, SET.ENABLE.RATE             JTR 12:30 10/30/85 )
 11 ( INIT.CTS                                   JTR 12:32 10/30/85 )
 12 ( WAIT, MSEC                                 JTR 10:58 04/08/88 )
 13 ( SEND.COM                                   JTR 13:20 02/24/88 )
 14 ( FDOS.EX                                    JTR 11:37 12/26/88 )
 15 ( Set-up Interrupt                           JTR 11:05 04/08/88 )
 16 ( @IMR, !IMR, RESET. MASK of 8259A           JTR 10:58 04/08/88 )
 17 ( RESET.MASK, MASK  8259A                    JTR 09:41 02/25/88 )
 18 ( Interrupt Service  1                       JTR 15:54 03/20/86 )
 19 ( Interrupt Service Cont.  2                 JTR 11:58 02/12/86 )
 20 ( Interrupt Service Cont. 3                  JTR 15:55 03/20/86 )
 21 (                                                16:20 01/12/86 )
 22 ( SET.BUFFERS                                JTR 10:59 04/08/88 )
ok
```

```
 23 ( Interrupt test                             JTR 10:59 04/08/88 )
 24 ( Display and Command Table Variables        JTR 10:59 04/08/88
 25 ( @DATA,                                     JTR 09:41 02/25/88 )
 26 ( @LAT.DATA                                  JTR 14:53 02/24/88 )
 27 ( .DATA                                      JTR 12:47 02/25/88 )
 28 ( .DATE/TIME                                 JTR 11:02 04/08/88 )
 29 ( DISPLAY , COM.LINE                         JTR 12:49 02/25/88 )
 30 ( File Storage Support                       JTR 13:43 02/12/92 )
 31 ( Storage support                            JTR 09:41 02/25/88 )
 32 ( Storage support                            JTR 13:56 02/12/92 )
 33 ( FILE.HEAD                                  JTR 13:15 02/25/86 )
 34 ( STORE.BUF                                  JTR 11:01 04/08/88 )
 35 ( STORE.BUFF                                 JTR 09:43 02/25/88 )
 36 ( TABLE Support                              JTR 11:02 04/08/88 )
 37 ( SERVICE TABLE                              JTR 11:02 04/08/88 )
 38 ( Table Commands                             JTR 09:43 02/25/88 )
 39 ( COMMAND.TABLE                              JTR 09:43 02/25/88 )
 40 ( Service Command table                      JTR 11:03 04/08/88 )
 41 ( KEY Functions                              JTR 13:58 02/12/92 )
 42 ( RUN                                        JTR 11:04 04/08/88 )
 43
 44
ok
```

```
Screen # 0
( CALIBRATION & TESTING SYSTEM  CTSD         JTR 11:37 04/08/88 )
( Last change:   Screen  019                 JTR 11:37 04/08/88 )

APPENDIX D
                    DUAL INSTRUMENT SOFTWARE
                         VERSION 1.1

Copyright (c) 1985
Research Support Instruments, Inc.
10610 Beaver Dam Road
Cockeysville, Maryland  21030

Written by:
     Tom Riley

FILE:  CTSD.SCR


Screen # 1
( Variables                                  JTR 11:06 04/08/88 )
: MARK ; ASM86 DOSINT ( Marks start of application words )
VARIABLE  DATA.RATE  ( Data clock rate in DecaHz, 10's )
VARIABLE  ENABLE.RATE   ( Enable Rate in Hz )
VARIABLE  DATA.BITS   ( Number of data bits )
VARIABLE  CUR.COM    0 CUR.COM !        ( Current Command)
VARIABLE  DATA.BYTES  ( Number of data Bytes )
     HEX
VARIABLE  DATA.POINT          ( Data Pointer )
4000 CONSTANT BUF#1.DS        ( Buffer #1 DS Register )
4800 CONSTANT BUF#2.DS        ( Buffer #2 DS Register )
VARIABLE  BUF.LEN             ( Buffer Length )
VARIABLE  NEX.BUF             ( Next Buffer DS )
VARIABLE  PRES.BUF            ( Present Buffer DS )
VARIABLE  NEX.BUF.STOR -->    ( Next Buffer to Disk DS )
     DECIMAL -->


Screen # 2
( INIT.VAR. CTS.ADDR                         JTR 11:07 04/08/88 )
DECIMAL
: INIT.VAR   (  --  , Initialize application variables )
        DECIMAL                ( Change initial values here )
        1000    DATA.RATE !        ( DecaHz. 10,000 Hz )
        10      ENABLE.RATE !      ( Hz  )
        96      DATA.BITS !        ( bits )
         2  1024 * DATA.BITS @ - 1 -    BUF.LEN !    ( Bytes )
        DATA.BITS @ 15 + 16 /  2 * DATA.BYTES !  ;
                 ( divisiable by 8, even number )
HEX
300  CONSTANT  CTS.ADDR    ( Base address of CTS card )
3    CONSTANT  INTERRUPT#      ( Interrupt Number )
DECIMAL -->
```

```
Screen # 3
( Store Data Rate Parameter                    JTR 13:54 03/07/88 )
HEX

CODE  DR!    ( N -- , Store Data Rate Parameter on card )
        DX, # CTS.ADDR 3 + MOV          ( Clock Mode Port Address)
        AL, # 36          MOV  ( Set Mode, Clk 0 Sq Wav)
        DX,  AL           OUT
        DX, # CTS.ADDR      MOV          ( Parameter Port )
        AX       POP                   ( get n from stack)
        DX,    AL         OUT          ( LSB )
        AL,    AH         MOV         ( Next BYTE)
        DX,    AL         OUT         ( MSB )
        NEXT, END-CODE      -->
```

```
Screen # 4
( Store Enable Rate Parameter                  JTR 09:40 02/25/88 )

CODE  ER!    ( N -- , Store Enable Rate Parameter on card )
        DX, # CTS.ADDR 3 + MOV          ( Clock Mode Port Address)
        AL, # 74          MOV  ( Set Mode, CLK 1 RATE GEN )
        DX, AL            OUT
        DX, # CTS.ADDR 1+  MOV          ( Parameter Port )
        AX       POP                   ( get n from stack)
        DX,    AL         OUT          ( LSB )
        AL, AH  MOV   DX, AL OUT        ( MSB)
        NEXT, END-CODE            -->
```

```
Screen # 5
( Store Bit Count Parameter                    JTR 09:40 02/25/88 )

CODE  BN!    ( N -- , Store Number of Bits on card )
        DX, # CTS.ADDR 3 + MOV          ( Clock Mode Port Address)
        AL, # B4          MOV  ( Set Mode, CLK 2 RATE GEN )
        DX, AL            OUT
        DX, # CTS.ADDR 2 + MOV          ( Parameter Port )
        AX                POP          ( get n from stack)
        DX,    AL         OUT          ( LSB )
        AL.    AH         MOV         ( Next BYTE)
        DX,    AL         OUT          ( MSB )
        NEXT, END-CODE

  -->
```

```
Screen # 6
( Start Data Rate, Reset Data Rate            JTR 11:07 04/08/88 )

CODE  START.DR    (  -- , Start Data Rate)
        DX, # CTS.ADDR 8 +  MOV     ( Port Address )
        AL, # 00            MOV     ( Any number )
        DX,   AL            OUT     ( write to address)
        NEXT, END-CODE

CODE  STOP.DR    (  -- , Reset Data Rate)
        DX, # CTS.ADDR 4 + MOV     ( Port Address )
        AL, # 00            MOV     ( Any number )
        DX,   AL            OUT     ( write to address)
        NEXT, END-CODE


             -->


Screen # 7
( Reset Enable, Send Command                   JTR 11:07 04/08/88 )

CODE  RESET.E    (  -- , Reset Enable)
        DX, # CTS.ADDR C + MOV     ( Port Address )
        AL, # 00            MOV     ( Any number )
        DX.   AL            OUT     ( write to address)
        NEXT. END-CODE

CODE  COMMAND!    ( N  -- , Issue Command  )
        AX      POP                  ( get n from stack)
        DX, # CTS.ADDR 10 +   MOV    ( Port Address )
        DX.   AL            OUT     ( write to address)
        NEXT, END-CODE

      -->


Screen # 8
( Read FIFO, Shift FIFO                        JTR 11:08 04/08/88 )

CODE  @FIFO   (  -- N , Read FIFO)
        DX, #  CTS.ADDR 14 + MOV     ( Port Address )
        AL,  DX              IN     ( Read 1 Byte )
        AH, # 0              MOV
        AX                   PUSH
        NEXT, END-CODE

CODE  SHIFT.FIFO   (  -- , Parallel Shift FIFO for next byte )
        DX, #  CTS.ADDR 18 + MOV     ( Port Address )
        AL, # 00            MOV     ( Any number )
        DX,   AL            OUT     ( write to address)
        NEXT, END-CODE

                          -->
```

```
Screen # 9
( Clear FIFO                                        JTR 11:46 10/30/85 )

CODE  CLEAR.FIFO    (  -- , Clear FIFO completely )
        DX, # CTS.ADDR 1C + MOV      ( Port Address )
        AL, # 00            MOV     ( Anynumber )
        DX,   AL            OUT     ( write to address)
        NEXT, END-CODE


DECIMAL
               -->


Screen # 10
( SET.DATA.RATE, SET.ENABLE.RATE                    JTR 12:30 10/30/85 )

: SET.DATA.RATE   ( N -- , Set Data Rate )
        2385 SWAP 100 SWAP  */  ( convert to parameter )
        DR! ;

: SET.ENABLE.RATE   ( N --   , Set Enable Rate )
        DATA.RATE @ 10 4 */
        SWAP /
        ER! ;

: SET.DATA.BITS   ( N  --   , Set number of data bits )
        DUP 256 > IF DROP 256 THEN      ( less than 256)
        BN! ;                 ( store )
-->


Screen # 11
( INIT.CTS                                          JTR 12:32 10/30/85 )

: INIT.CTS    (   --   , Initiate CTS card )
        INIT.VAR
        STOP.DR
        RESET.E
        0 COMMAND!
        CLEAR.FIFO
        DATA.RATE @  SET.DATA.RATE
        ENABLE.RATE @  SET.ENABLE.RATE
        DATA.BITS @  SET.DATA.BITS
          ;


   -->
```

```
Screen # 12
( WAIT, MSEC                                    JTR 11:08 04/08/88 )

: WAIT   ( N --   , Short delay )
        0 DO LOOP ;

: MSEC    ( N -- , delay in milliseconds )
        0 DO  36 WAIT  LOOP ;

 -->
```

```
Screen # 13
( SEND.COM                                      JTR 13:20 02/24/88 )

: SEND.COM   ( N --   . Send CTS command as Level )
        DUP 8 > IF DROP 0 THEN    ( Less than 8)
        DUP 1 < IF DROP 0 THEN    ( Greater than 1 )
        DUP  CUR.COM !            ( Current command )
        2 SWAP BEGIN DUP WHILE
             1- SWAP 2* SWAP REPEAT        ( Shift to bit )
        DROP 2/ 2/   COMMAND! ;
: PULSE.COM   ( N --   , Send CTS command as Pulse 0 )
        DUP CUR.COM !            ( Update Current Command )
        DUP SEND.COM             ( Set Bit)
        100 MSEC                  ( wait )
    (   0  SEND.COM   )          ( 0 Bits )
        CUR.COM !                ( Store last command )
        ; -->
```

```
Screen # 14
( FDOS.EX                                       JTR 11:37 12/26/88 )
HEX

CODE FDOS.EX   ( AX, DS, DX -- AX, BX , Extended FDOS )
        DX POP DS POP AX POP    ( Load Parameters )
        SI PUSH BP PUSH          ( Conserve pointers )
        21 INT                   ( Call DOS Interrupt )
        BP POP SI POP            ( Restore Pointers )
        AX PUSH BX PUSH          ( Return results )
        NEXT, END-CODE



  -->
```

```
Screen # 15
( Set-up Interrupt                                JTR 11:16 04/08/88 )

CODE @CS     (    --  N , Fetch CS register )
        CS PUSH    NEXT, END-CODE

CODE @DS     (    --  N , Fetch DS register )
        DS PUSH    NEXT, END-CODE

CODE SOFT.INT  (      -- , Call Interrupt from Software )
        13      INT
        NEXT, END-CODE

-->
```

```
Screen # 16
( @IMR, !IMR, RESET. MASK of 8259A          JTR 11:08 04/08/88 )
HEX
CODE @IMR  (  -- N, Read Interrupt Mask Register of 8259A )
        # 21 , AL  IN
        AH, # 0 MOV
        AX PUSH
        NEXT, END-CODE
CODE !IMR  ( N -- , Store Interrupt Mask Register )
        AX   POP
        # 21 , AL  OUT  STI  ( Enable interrupt )
        NEXT, END-CODE




-->
```

```
Screen # 17
( RESET.MASK, MASK  8259A                       JTR 09:41 02/25/88 )

: RESET.MASK  (  -- , Reset 8259 Interrupt Mask )
        @IMR                            ( Read Mask )
        1 INTERRUPT# SHIFT   NOT        ( 0 in correct bit )
        AND !IMR ;                      ( Store Mask )



: MASK  (  -- , Mask 8259 Interrupt Mask )
        @IMR                            ( Read Mask )
        1 INTERRUPT# SHIFT              ( 1 in correct bit )
        OR !IMR ;                       ( Store Mask )


          -->
```

Laboratory Microsystems PC/FORTH 3.00   13:28   10/30/89   a:ctsd.scr

```
Screen # 18
( Interrupt Service  1                          JTR 15:54 03/20/86 )
HEX
CODE INTER.SERV ( -- , Assembly Interrupt Service Routine )
        PUSHF AX PUSH  BX PUSH  CX PUSH  DX PUSH  ( Save   )
        BP PUSH SI PUSH DS PUSH  DS PUSH ( System Configure )
        AX,  CS  MOV   DS,  AX  MOV ( CS to DS )
        AX,   DATA.BYTES MOV  CX,  AX MOV ( Count Data Bytes )

1$:     AX, DATA.POINT  MOV  BX, AX  MOV  ( Data Pointer )
        AX, PRES.BUF  MOV   DS,  AX  MOV  ( Data Segment )
        DX, # CTS.ADDR 14 + MOV   ( READ FIFO )
        AL,  DX          ,    IN   ( Read FIFO )
        [BX],  AL     MOV    ( Store Byte )
        DX, # CTS.ADDR 18 +  MOV  ( Shift FIFO )
        DX,  AL              OUT
      -->


Screen # 19
( Interrupt Service Cont. 2                     JTR 11:37 04/08/88 )
        AX,  CS  MOV   DS,  AX  MOV ( CS to DS )
        WORD  DATA.POINT  INC      ( Increment Data Pointer )
        CX      DEC               ( Reduce count )
        AX,  CX  MOV    AX,  AX  AND  ( Set flag)
        1$              JNZ

        DX,  # CTS.ALDR 1C +  MOV  ( CLEAR FIFO )
        DX,  AL              OUT  ( Clear FIFO )

        AX.  CS  MOV   DS,  AX  MOV ( CS to DS )
        AX,  BUF.LEN    MOV        ( End of Buffer )
        BX,  DATA.POINT MOV
        AX, BX          CMP
        2$      JNL
 -->


Screen # 20
( Interrupt Service Cont. 3                     JTR 15:55 03/20/86 )
        AX,     PRES.BUF MOV       ( Swap DS of Present )
        BX,     NEX.BUF  MOV       ( and Next Buffer )
        PRES.BUF , BX   MOV
        NEX.BUF ,  AX   MOV
        AX,  # 00       MOV     ( Zero Pointer )
        DATA.POINT , AX MOV
2$:     DX,  # CTS.ADDR  C + MOV  ( Reset Card Interrupt )
        DX,  AL        OUT        ( Any number out )
        DX, # 20    MOV  AL, # 20  MOV    ( Reset 8259A )
        DX,  AL   OUT

        DS POP  CS POP  SI POP BP POP
        DX POP  CX POP  BX POP  AX POP POPF   ( Restore )
        IRET  END-CODE             ( Return from Interrupt )
    DECIMAL  -->
```

```
Screen # 21
(                                               16:20 02/12/86 )









  -->




Screen # 22
( SET.BUFFERS                               JTR 11:09 04/08/88 )
HEX
: SET.INTER.VEC ( --   , Set Interrupt Vector )
        25 100 * INTERRUPT# 8 + +
        @CS ['] INTER.SERV 2 +       ( Sector and Code Addr)
       FDOS.EX  DROP DROP
           ;


: SET.BUFFERS  ( -- , Set-up Data Buffers )
        BUF#1.DS  PRES.BUF !
        BUF#2.DS  NEX.BUF !
        BUF#1.DS  NEX.BUF.STOR !
        0 DATA.POINT ! ;
     -->


Screen # 23
( Interrupt test                            JTR 11:09 04/08/88 )
HEX
: INIT.INTER ( -- , Initiate Interrupt Routine )
     SET.INTER.VEC                  ( Set-up Interrupt Vec )
     SET.BUFFERS                    ( Set-up Data Buffers )
     RESET.MASK                     ( Enable interrupt Mask )
     ;
DECIMAL

: TEST  ( N --   , INTERRUPT TEST )
        0 DO
                DATA.POINT @ U. PRES.BUF @ U. CR
        LOOP ;


    -->
```

```
Screen # 24
( Display and Command Table Variables        JTR 11:10 04/08/88 )

VARIABLE  ?DATA Ø ?DATA ! ( Data taking Flag Ø - Taking Data )
VARIABLE ?STORING  Ø ?STORING ! ( Storage Flag Ø - Storing )
VARIABLE STOR.POINT      ( Data storage memory pointer )
VARIABLE  FRAM.STOR  Ø FRAM.STOR ! ( Data Frames Stored )
VARIABLE   ELAPS.TIME      ( Elapsed time for table, sec )
VARIABLE   CUR.ENT         ( Current Entry from table )
VARIABLE   START.TIME      ( Start time for table, sec )

: TOG.DATA  (  -- , Toggle ?DATA )
        ?DATA @  NOT DUP ?DATA !     ( Set Flag )
        IF START.DR ELSE STOP.DR THEN ; ( Start/stop Data Rate)

: TOG.STOR ( --  ,  Toggle ?STORING )
        ?STORING @ NOT ?STORING ! ;   -->


Screen # 25
( @DATA.                                     JTR 09:41 02/25/88 )

CODE @DATA ( ADDR Seg -- N, Fetch Data from data buffer )
        DS      POP                   ( Get Segment )
        BX      POP                   ( Get Data Address )
        AX,     [BX]    MOV         ( Get 2 Bytes )
        BX,     AX      MOV         ( Save )
        AX,     CS      MOV         ( Restore DS )
        DS,     AX      MOV
        BX              PUSH        ( Place on Stack )
        NEXT, END-CODE

: ON/OFF  ( F --  , Print ON  or OFF )
        IF ." ON   " ELSE ." OFF " THEN ;
-->


Screen # 26
( @LAT.DATA                                  JTR 14:53 02/24/88 )

: @LAT.DATA (   -- Nn ... N1, Fetch Latest Data )
        DATA.BYTES @
        Ø  DO
               DATA.POINT @  I - 1 -    PRES.BUF @
               @DATA
           2 +LOOP ;

         -->
```

```
Screen # 27
( .DATA                                            JTR 14:02 03/07/88 )
: .DATA    ( --    , Display Latest Data )
  @LAT.DATA   ." Master Count          : " 8 U.R  CR
  ." Master Wavelength : " DUP 1023 AND 8 U.R  CR
  ." Master Flags       : " 256 / 255 AND 8 BINARY U.R DECIMAL
                                   CR
  ." Master Command     : " DUP 256 / 15 AND 8 U.R CR
                            SP@ 1+ SP!
  ." Platform Angle     : " DUP 8191 AND 8 U.R  CR
  ." Platform Flags     : " 4096 / 7 AND 8 BINARY U.R DECIMAL CR
  ." Slave Count        : " 8 U.R  CR
  ." Slave Wavelength   : " DUP 1023 AND 8 U.R  CR
  ." Slave Flags        : " 256 / 255 AND 8 BINARY U.R
                            DECIMAL CR
  ." Slave Command      : " DUP 256 / 15 AND 8 U.R CR CR
                               SP@ 1- SP!   DROP  ;       --)


Screen # 28
( .DATE/TIME                                       JTR 11:11 04/08/88 )

: .DATE/TIME ( -- , Print Date and time )
        @DATE  256 /MOD SWAP
             ." Day " . ." , Month " .
             ." , Year " . 8 SPACES
        @TIME SWAP 256 /MOD  .
        ." : " . 256 /MOD ." : " . DROP
        9 SPACES  CR ;

: .STATUS    (  --   , Status Display )
        CUR.COM @  ." Current Command - " . CR
        ?DATA @    ." Data Taking      - " ON/OFF 12 SPACES
        ?STORING @ ." Storing Data     - " ON/OFF  CR
        ELAPS.TIME @ ." Elapsed Time    - " 8 U.R 8 SPACES
        FRAM.STOR @ ." Frames Stored    - " U. 4 SPACES CR ; --)


Screen # 29
( DISPLAY . COM.LINE                               JTR 09:51 02/25/88 )

: DISPLAY  ( --   , Standard display )
        0 0 GOTOXY
        10 SPACES ." RSI    CALIBRATION AND TESTING " CR CR
        .DATE/TIME CR
        .DATA
        .STATUS  CR
 ." Active Keys - " CR
 ."    0 - 7  Send Command " CR
 ."    D -    Toggle Data Taking      T - Run Command Table" CR
 ."    S -    Toggle Data Storage     Q - Quit Program " CR :

: COM.LINE (   --    , Move cursor to clear command line )
      0 23 GOTOXY 60 SPACES 0 23 GOTOXY ;
--)
```

```
Screen # 30
( File Storage Support                          JTR 13:54 03/07/88 )

HANDLE FILE1
VARIABLE WBUF  100 ALLOT   ( Write Buffer )
VARIABLE WLEN  0 WLEN !     ( Length of WBUF text string )
: WRITE.LINE   (  --   , Write line from WBUF to disk )
        FILE1 WLEN @ DUP >R  WBUF  WRITE
        IF COM.LINE ." Write error " CR QUIT THEN
        R> <>
        IF CR ." Media Full " CR QUIT THEN
        0 WLEN ! ;
: ADD.SP   ( -- , Add space to WBUF )
     32 WBUF WLEN @ +   C!     ( ASCII Space 32 )
      1  WLEN  +! ;
                                                   -->


Screen # 31
( Storage support                               JTR 09:41 02/25/88 )

: ADD.NUM    ( N --   , Adds number to WBUF )
        0 <# #S #>
        0 DO  DUP C@ WBUF WLEN @ + C!
           1  WLEN  +! 1 + LOOP DROP ADD.SP ;
: ADD.CR   ( -- , Add CR and LF to WBUF )
        13 WBUF WLEN @ + DUP ROT ROT C! ( CR )
        1 + 10 SWAP C!                  ( LF )
        2 WLEN  +! ;              ( count )


: OPEN.IT        FILE1 OPEN-FILE
                 IF COM.LINE ." Can't open file" CR QUIT THEN ;

                                                        -->


Screen # 32
( Storage support                               JTR 13:56 02/12/92 )
: GET-FILENAME   ( -- , Enter File name )
                 COM.LINE
                 ." Enter Path and Filename:   "
                 FILE1 INPUT-FILENAME  CR
                 FILE1 MAKE-FILE DROP OPEN.IT ;

: CLOSE.IT       FILE1 CLOSE-FILE DROP ;

: ADD.DATE  (  --  , Add Date and Time to WBUF )
        ADD.CR @DATE SWAP ADD.NUM
        256 /MOD  ADD.NUM ADD.NUM
        @TIME SWAP 256 /MOD ADD.NUM ADD.NUM
        256 / ADD.NUM  ADD.CR  ADD.CR
        WRITE.LINE ;                             -->
```

```
Screen # 33
( FILE.HEAD                                     JTR 13:15 02/25/86 )

: FIND.LEN (  --    , Find Length of WBUF )
        0 WLEN ! WBUF 0 BEGIN
               2DUP  + C@ WHILE
               1 WLEN +! 1 + REPEAT DROP DROP ;

: FILE.HEAD   (  --   , Get file Heading   )
        COM.LINE  ." Enter File Heading : "
        WBUF 64 DUP WLEN ! EXPECT      ( Get Text String )
        FIND.LEN ADD.CR ADD.CR         ( Number of Char )
        WRITE.LINE ;


-->


Screen # 34
( STORE.BUF                                     JTR 11:12 04/08/88 )

: STORE.BUF    ( --   , Store buffer on disk )
        0 STOR.POINT !     ADD.DATE
        BUF.LEN @ 0 DO                 ( Entire buffer )
            FRAM.STOR @ ADD.NUM
            1 FRAM.STOR +!             ( Frame count )
            DATA.BYTES @ 0 DO          ( One data Frame )
              NEX.BUF.STOR @ STOR.POINT @ @L   ( Get Datum )
              ADD.NUM                 ( Add to WBUF )
              2 STOR.POINT  +!
              2 +LOOP ADD.CR              ( Add CR )
            WRITE.LINE
        DATA.BYTES @ +LOOP             ( Next Data Frame )
        PRES.BUF @   NEX.BUF.STOR ! ;
-->


Screen # 35
( STORE.BUFF                                    JTR 09:43 02/25/88 )

: ?STORE.BUFF (  --   , Store Buffer to Disk )
        ?STORING @                          ( Storing ? )
           NEX.BUF @  NEX.BUF.STOR @ = AND   ( Buf Full ? )
               IF    STORE.BUF            ( Store buffer )
        THEN ;



: STOP   (   --  , Stop the program )
        CLOSE.IT  STOP.DR MASK COM.LINE ;

                                         -->
```

```
Screen # 36
( TABLE Support                                    JTR 11:13 04/08/88 )

: ENTER   (  N ADDR --   , Make entry into table )
         SWAP , , ;    ( Move address and time into memory )

: @SEC   (  -- N, Get number of seconds today )
        @TIME 256 /MOD SWAP DROP SWAP    ( Get seconds )
           256 /MOD 6 - 60 * + 60 * + ; ( Hours and minutes )

: !START.TIME   (  --   , Get start time )
        @SEC  START.TIME ! ;

: @ELAPS.TIME (  -- N , Get elapsed time )
        @SEC START.TIME @ - DUP ELAPS.TIME ! ;

-->

Screen # 37
( SERVICE TABLE                                    JTR 09:43 02/25/88 )

: RESTART.TABLE ( -- , Restart Table )
        !START.TIME  @ELASP.TIME DROP 0  CUR.ENT ! ;
: END.TABLE   (  --   , End the Table )
        0  ['] RESTART.TABLE  ENTER ;    ( Last entry )

                        -->

Screen # 38
( Table Commands                                   JTR 09:43 02/25/88 )


: STORE ( --   ,  Start Data Storing )
        -1 ?STORING ! ;

: COM.#1 ( -- ,  Issue Command # 1 )
        1 PULSE.COM ;
: COM.#0 ( -- ,  Issue Command # 0 )
        0 PULSE.COM ;



-->
```

```
Screen # 39
( COMMAND.TABLE                                    JTR 09:43 02/25/88 )

VARIABLE  COMMAND.TABLE    -2 ALLOT    ( Memory Heading )
( Time     '   Command       ENTER       )
  10       '   COM.#1        ENTER
  20       '   COM.#0        ENTER
  30       '   COM.#1        ENTER
  40       '   COM.#0        ENTER


END.TABLE -->
```

```
Screen # 40
( Service Command table                            JTR 11:14 04/08/88 )

: TABLE.EXEC   ( N-- , Execute from Table )
        COMMAND.TABLE + 2 + @      ( Calculate address )
             EXECUTE               ( Execute the command )
        CUR.ENT @ 4 + CUR.ENT ! ; ( Iterate Current Entry )
: TIME.CHECK   ( -- F or N , T , Compare Time )
        COMMAND.TABLE CUR.ENT @ + @  @ELAPS.TIME  <=
        IF   CUR.ENT @ -1  ELSE 0 THEN ;
: COMMANDS ( --    ,   Run Command Table )
      RESTART.TABLE BEGIN  ?TERMINAL NOT WHILE
          TIME.CHECK  IF TABLE.EXEC   ( time for next com ? )
                   ELSE DISPLAY  ( Refresh display )
                        ?STORE.BUFF
          THEN REPEAT
          KEY DROP 0 ELAPS.TIME ! ; ( Any key to stop ) -->
```

```
Screen # 41
( KEY Functions                                    JTR 13:58 02/12/92 )

: SERV.KEY     ( -- , Service Keyboard )
        ?TERMINAL    IF KEY          ( Key pressed ? )
           DUP DUP ASCII 0 )= SWAP ASCII 7 <= AND   ( 0 - 7 ? )
                IF DUP ASCII 0 - PULSE.COM THEN  ( Send Com )
           DUP ASCII D  = IF                    ( D ? )
                TOG.DATA  THEN                   ( Set flag )
           DUP ASCII S  = IF                    ( S )
                TOG.STOR  THEN                   ( Set flag )
           DUP ASCII T  = IF                    ( T )
                COMMANDS  THEN                   ( Run Table )
           DUP ASCII Q = IF
               STOP QUIT   THEN                  ( Quit )
           DROP THEN ;
-->
```

Screen # 42

```
( RUN                                          JTR 11:15 04/08/88 )

: RUN    ( -- , Run test )
        INIT.CTS                         ( Initiate system )
        INIT.INTER
        CLEARSCREEN  DISPLAY             ( Set-up Monitor )
        GET-FILENAME FILE.HEAD           ( Set-up Disk File )
        BEGIN  1 WHILE                   ( Infinite loop )
           DISPLAY
           SERV.KEY
           ?STORE.BUFF
        REPEAT ;                   ( Exit on key Q )
```

Screen # 43

Screen # 44

```
 0 ( CALIBRATION & TESTING SYSTEM   CTSD       JTR 11:37 04/08/89 )
 1 ( Variables                                 JTR 11:06 04/08/88 )
 2 ( INIT.VAR, CTS.ADDR                        JTR 11:07 04/08/88 )
 3 ( Store Data Rate Parameter                 JTR 13:54 03/07/88 )
 4 ( Store Enable Rate Parameter               JTR 09:40 02/25/88 )
 5 ( Store Bit Count Parameter                 JTR 09:40 02/25/88 )
 6 ( Start Data Rate, Reset Data  Rate         JTR 11:07 04/08/88 )
 7 ( Reset Enable, Send Command                JTR 11:07 04/08/88 )
 8 ( Read FIFO, Shift FIFO                     JTR 11:08 04/08/88 )
 9 ( Clear FIFO                                JTR 11:46 10/30/85 )
10 ( SET.DATA.RATE, SET.ENABLE.RATE            JTR 12:30 10/30/85 )
11 ( INIT.CTS                                  JTR 12:32 10/30/85 )
12 ( WAIT, MSEC                                JTR 11:08 04/08/88 )
13 ( SEND.COM                                  JTR 13:20 02/24/88 )
14 ( FDOS.EX                                   JTR 11:37 12/26/88 )
15 ( Set-up Interrupt                          JTR 11:16 04/08/88 )
16 ( @IMR, !IMR, RESET. MASK of 8259A          JTR 11:08 04/08/88 )
17 ( RESET.MASK, MASK  8259A                   JTR 09:41 02/25/88 )
18 ( Interrupt Service  1                      JTR 15:54 03/20/86 )
19 ( Interrupt Service Cont. 2                 JTR 11:37 42/08/88 )
20 ( Interrupt Service Cont. 3                 JTR 15:55 03/20/86 )
21 (                                               16:20 02/12/86 )
22 ( SET.BUFFERS                               JTR 11:09 04/08/88 )
ok


23 ( Interrupt test                            JTR 11:09 04/08/88 )
24 ( Display and Command Table Variables       JTR 11:10 04/08/88
25 ( @DATA,                                    JTR 09:41 02/25/88 )
26 ( @LAT.DATA                                 JTR 14:53 02/24/88 )
27 ( .DATA                                     JTR 14:02 03/07/88 )
28 ( .DATE/TIME                                JTR 11:11 04/08/88 )
29 ( DISPLAY , COM.LINE                        JTR 09:51 02/25/88 )
30 ( File Storage Support                      JTR 13:54 03/07/88 )
31 ( Storage support                           JTR 09:41 02/25/88 )
32 ( Storage support                           JTR 13:56 02/12/92 )
33 ( FILE.HEAD                                 JTR 13:15 02/25/86 )
34 ( STORE.BUF                                 JTR 11:12 04/08/88 )
35 ( STORE.BUFF                                JTR 09:43 02/25/88 )
36 ( TABLE Support                             JTR 11:13 04/08/88 )
37 ( SERVICE TABLE                             JTR 09:43 02/25/88 )
38 ( Table Commands                            JTR 09:43 02/25/88 )
39 ( COMMAND.TABLE                             JTR 09:43 02/25/88 )
40 ( Service Command table                     JTR 11:14 04/08/88 )
41 ( KEY Functions                             JTR 13:58 02/12/92 )
42 ( RUN                                       JTR 11:15 04/08/88 )
43
44
ok
```

FIGURES 1 THROUGH 15

HORIZON ULTRAVIOLET PROGRAM (HUP)

AFGL 801A

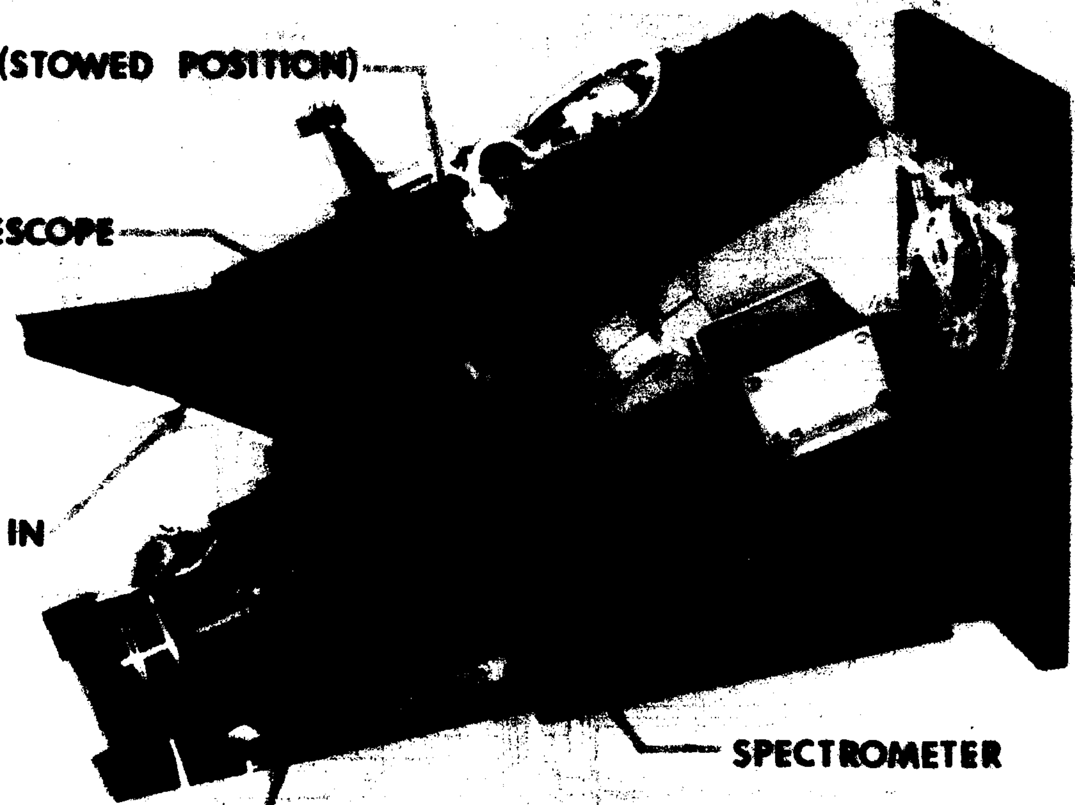
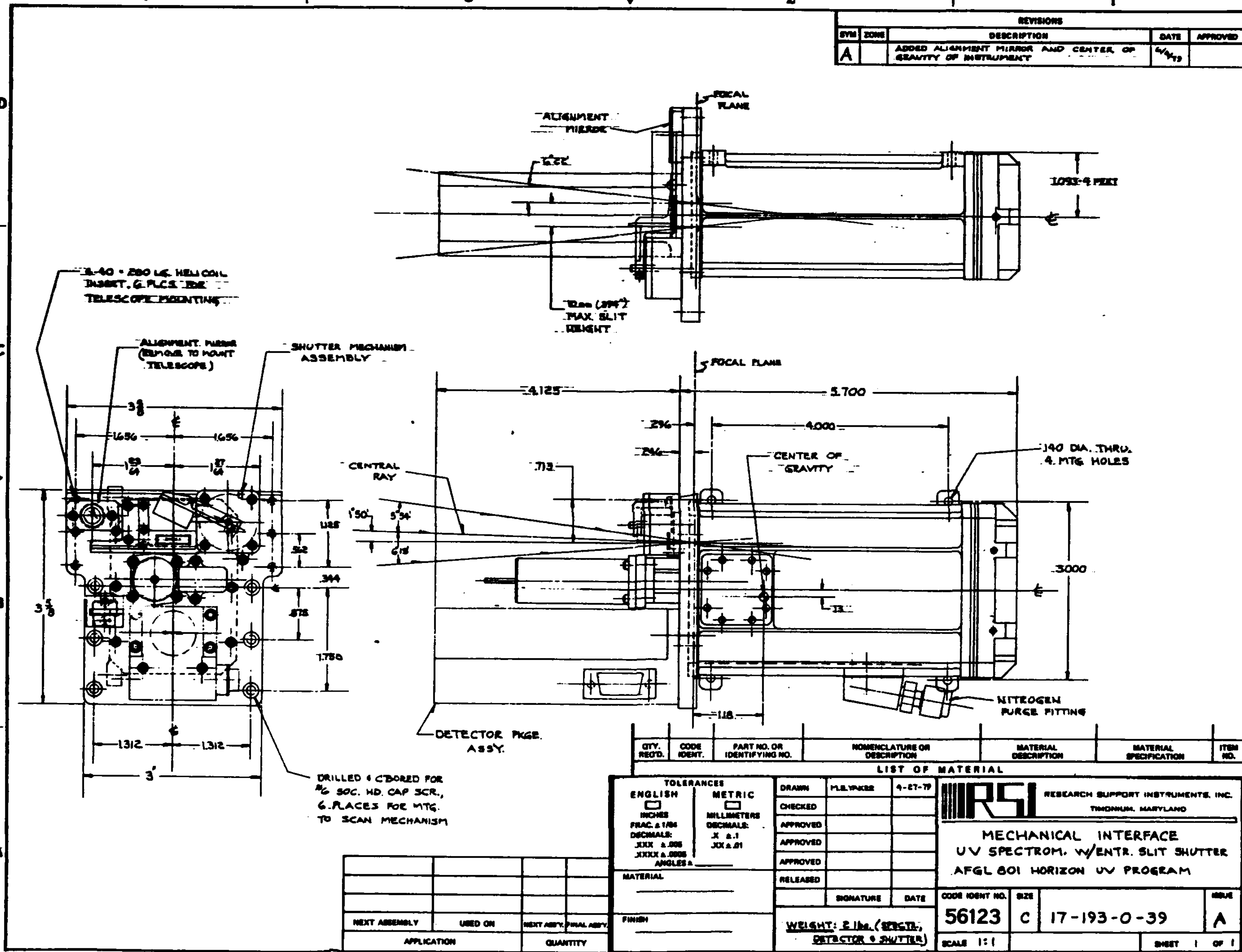


Fig. 1

SHUTTLE INSTRUMENT



REVISIONS
SYM
ZONE
DESCRIPTION
DATE
APPROVED
A
ADDED ALIGNMENT MIRROR AND CENTER OF GRAVITY OF INSTRUMENT
FOCAL PLANE
ALIGNMENT MIRROR
1.093-4 PERT
MAX SLIT HEIGHT
8-40 • 280 LG. HELI COIL INSERT, 6 PLCS. FOR TELESCOPE MOUNTING
ALIGNMENT MIRROR (REMOVE TO MOUNT TELESCOPE)
SHUTTER MECHANISM ASSEMBLY
FOCAL PLANE
4.125
5.700
.296
4.000
CENTER OF GRAVITY
.140 DIA. THRU. 4. MTG HOLES
CENTRAL RAY
.713
3.000
NITROGEN PURGE FITTING
DETECTOR PKGE. ASSY.
1.656
1.312
3'
1.750
.344
.875
.512
1.125
DRILLED & C'BORED FOR #6 SOC. HD. CAP SCR., 6 PLACES FOR MTG. TO SCAN MECHANISM
QTY. REQ'D.
CODE IDENT.
PART NO. OR IDENTIFYING NO.
NOMENCLATURE OR DESCRIPTION
MATERIAL DESCRIPTION
MATERIAL SPECIFICATION
ITEM NO.
LIST OF MATERIAL
TOLERANCES
ENGLISH
INCHES
METRIC
MILLIMETERS
DECIMALS
ANGLES
MATERIAL
FINISH
DRAWN
M.E.YAKER
4-27-79
CHECKED
APPROVED
RELEASED
SIGNATURE
DATE
RSI
RESEARCH SUPPORT INSTRUMENTS, INC.
TIMONIUM, MARYLAND
MECHANICAL INTERFACE
UV SPECTROM. W/ENTR. SLIT SHUTTER
AFGL 801 HORIZON UV PROGRAM
CODE IDENT NO.
SIZE
56123
C
17-193-0-39
ISSUE
A
SCALE 1:1
SHEET 1 OF 1
WEIGHT: 2 lbs. (SPECTR., DETECTOR & SHUTTER)
NEXT ASSEMBLY
USED ON
APPLICATION
QUANTITY

Fig. 2

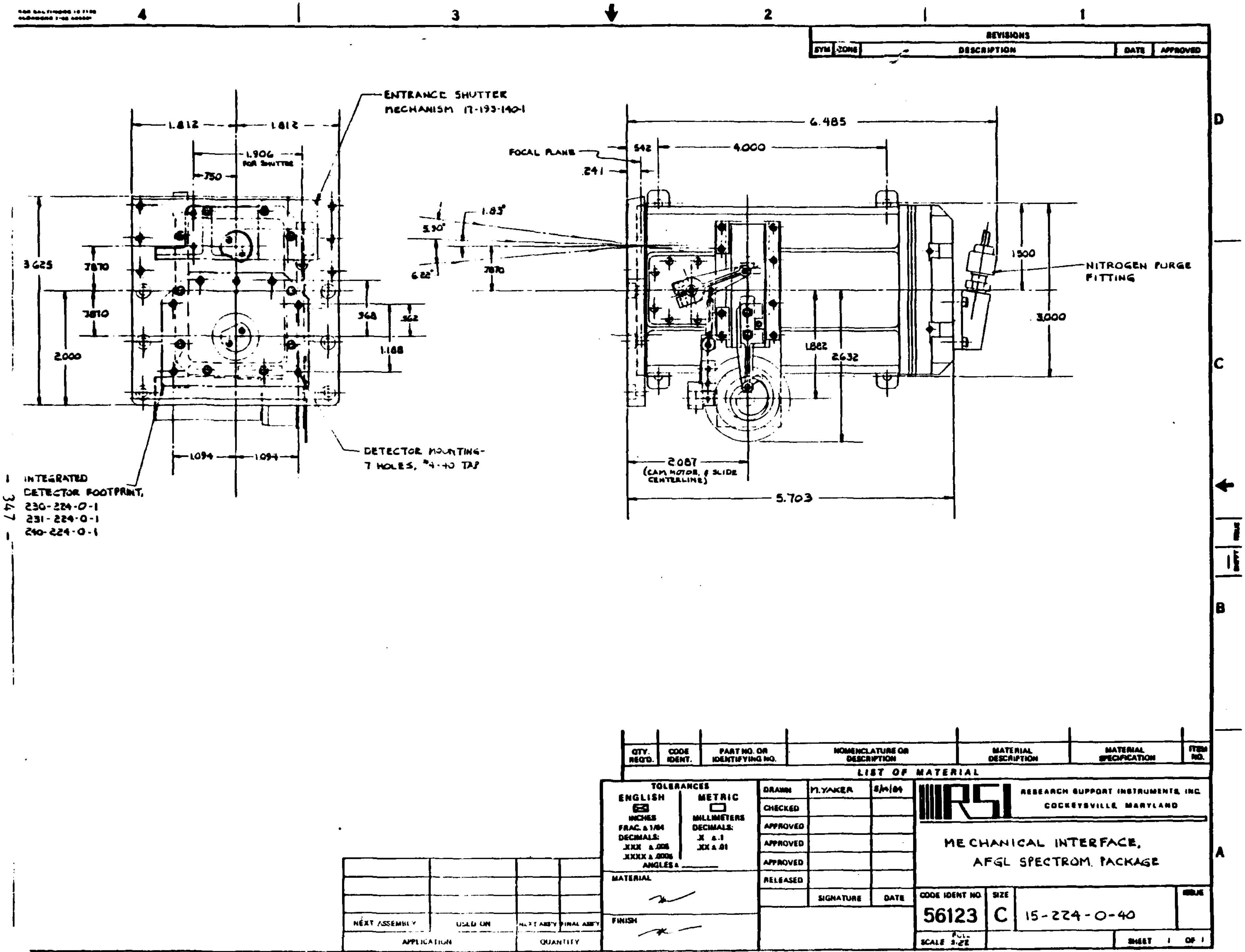
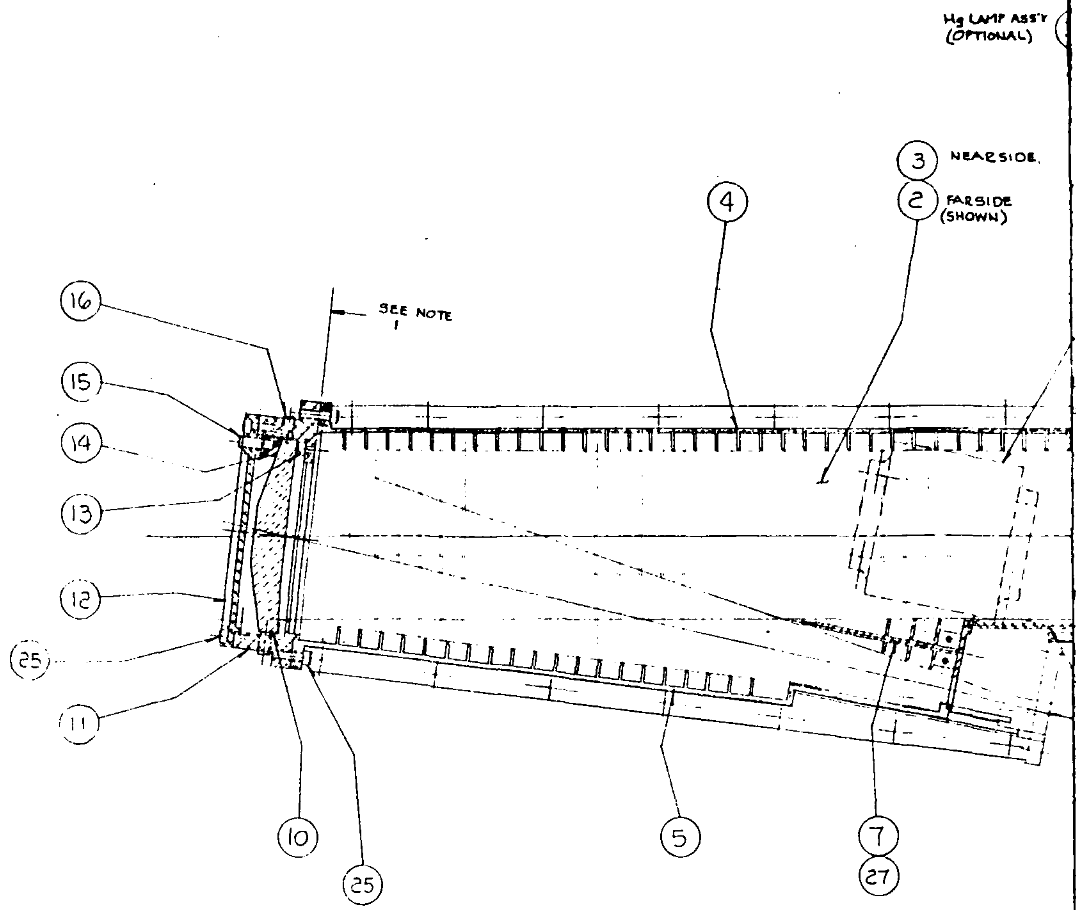

REVISIONS
SYM ZONE DESCRIPTION DATE APPROVED
ENTRANCE SHUTTER
MECHANISM 17-193-140-1
1.812
1.812
1.906
FOR SHUTTER
.750
3.625
.7870
.7870
2.000
.368
.562
1.188
1.094
1.094
DETECTOR MOUNTING-
7 HOLES, #4-40 TAP
INTEGRATED
DETECTOR FOOTPRINT,
230-224-0-1
231-224-0-1
240-224-0-1
FOCAL PLANE
.542
.241
4.000
6.485
1.83°
5.30°
6.22°
.7870
1.500
3.000
NITROGEN PURGE
FITTING
1.882
2.632
2.087
(CAM MOTOR, ℄ SLIDE
CENTERLINE)
5.703
QTY. REQ'D.
CODE IDENT.
PART NO. OR IDENTIFYING NO.
NOMENCLATURE OR DESCRIPTION
MATERIAL DESCRIPTION
MATERIAL SPECIFICATION
ITEM NO.
LIST OF MATERIAL
TOLERANCES
ENGLISH
INCHES
FRAC. ± 1/64
DECIMALS:
.XXX ± .005
.XXXX ± .0005
ANGLES ±
METRIC
MILLIMETERS
DECIMALS:
.X ± .1
.XX ± .01
MATERIAL
FINISH
DRAWN M. YAKER 8/14/84
CHECKED
APPROVED
APPROVED
APPROVED
RELEASED
SIGNATURE
DATE
RSI
RESEARCH SUPPORT INSTRUMENTS, INC.
COCKEYSVILLE, MARYLAND
MECHANICAL INTERFACE,
AFGL SPECTROM. PACKAGE
CODE IDENT NO.
56123
SIZE
C
15-224-0-40
ISSUE
SCALE FULL SIZE
SHEET 1 OF 1
NEXT ASSEMBLY
USED ON
NEXT ASSY
FINAL ASSY
APPLICATION
QUANTITY

Fig. 3



NOTES-

1)- PREASSEMBLE ITEMS 2, 3, 4, 5 & 6 WITH #0-80 SCREWS IN RIVET HOLES AND FACE-OFF ENDS (.010")

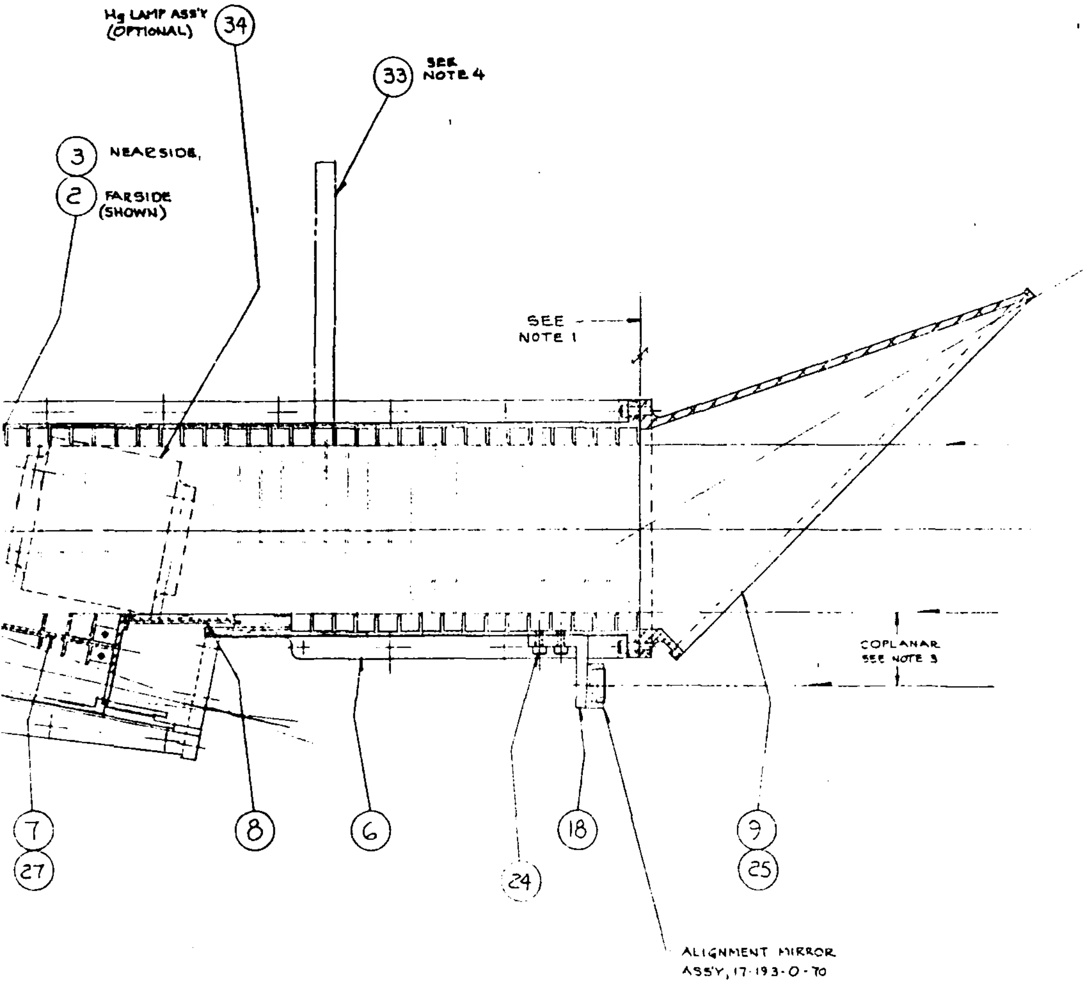
2)- ASSEMBLE USING 1/16 DIA. RIVETS AFTER ALL MACHINING AND BLACK ANODIZING IS COMPLETE.

3)- MIRROR MOUNTING SURFACE OF BRACKET TO BE CUSTOM FITTED SO THAT ALIGNMENT MIRROR ASS'Y (17-193-0-70) WILL MOUNT WITH MIRROR NORMAL COPLANAR TO CENTRAL ENTRANCE RAY WITHIN ONE MINUTE (0°-1') OF ARC.

4)- IF ITEM 33 (DUST COVER, 140-193-140-1) IS USED, SEE DRAWING 140-193-0-23 FOR CORRECT MODIFICATIONS TO UPPER PLATE, 140-193-0-4.

NEXT AS

SYM	ZONE	REVISIONS DESCRIPTION	DATE	APPROVED



TH #0-80
OFF ENDS

TER ALL
S COMPLETE.
T TO BE CUSTOM
SS'Y (17-193-0-70)
PLANAR TO CENTRAL
(0°-1') OF ARC
) IS USED, SEE
MODIFICATIONS

QTY. REQ'D.	CODE IDENT.	PART NO. OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION	MATERIAL DESCRIPTION	MATERIAL SPECIFICATION	ITEM NO

LIST OF MATERIAL

NEXT ASSEMBLY	USED ON	NEXT ASS'Y	FINAL ASS'Y
APPLICATION		QUANTITY	

TOLERANCES
ENGLISH — INCHES — FRAC. ± 1/64 — DECIMALS: .XXX ± .005, .XXXX ± .0005 — ANGLES ±
METRIC — MILLIMETERS — DECIMALS: .X ± .1, .XX ± .01

MATERIAL

FINISH
SEE NOTE 2

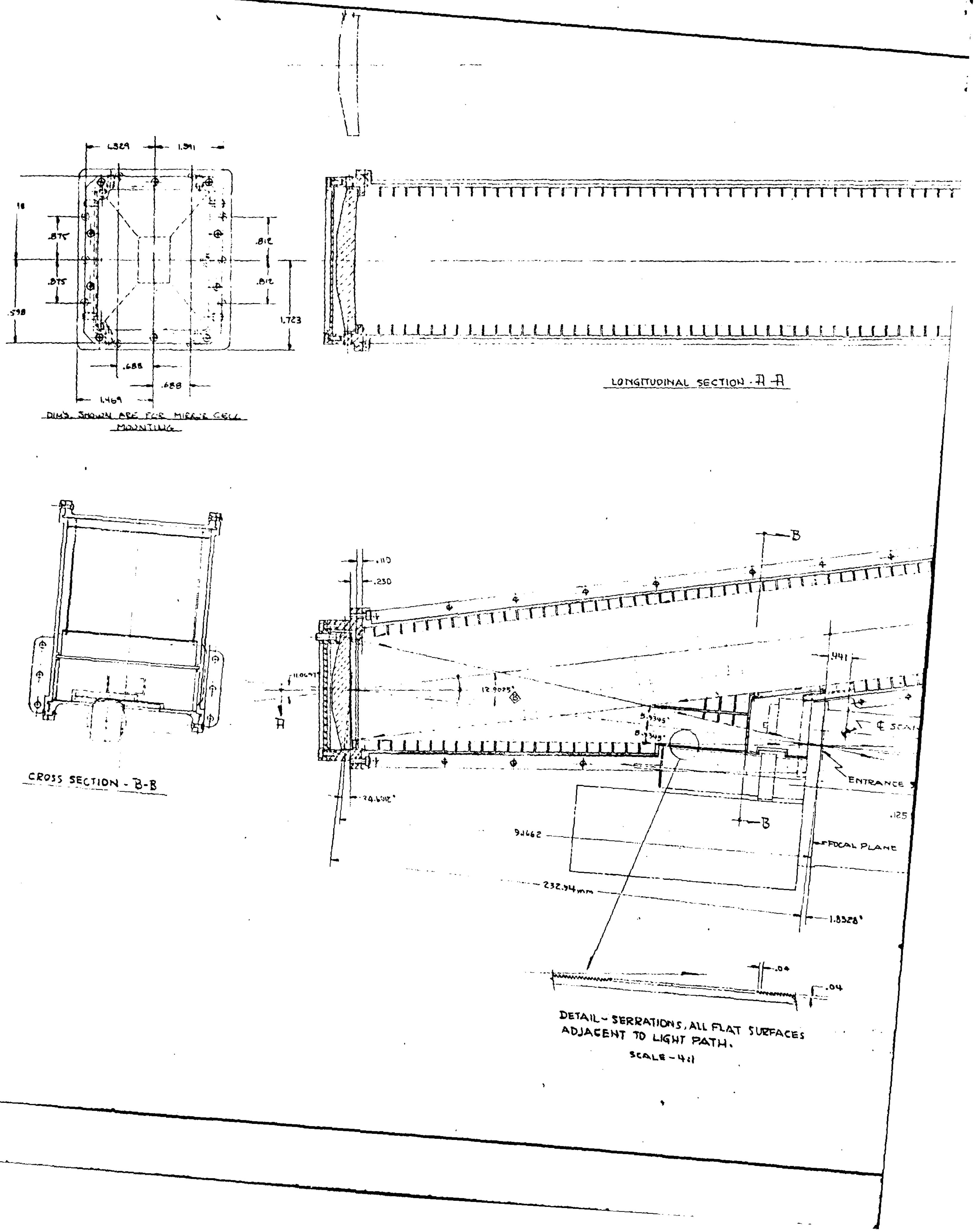
	SIGNATURE	DATE
DRAWN	ME YAKER	1-22-80
CHECKED		
APPROVED		
APPROVED		
APPROVED		
RELEASED		

RSI RESEARCH SUPPORT INSTRUMENTS, INC.
TIMONIUM, MARYLAND

ENTRANCE TELESCOPE ASSEMBLY
AFGL 801
HIRIS UV PROGRAM

CODE IDENT NO	SIZE		ISSUE
56123	D	140-193-0-1	D
SCALE 1:1		SHEET 1 OF 2	

Fig. 4


1.329
1.591
.875
.875
.598
.812
.812
1.723
.688
.688
1.469
DIMS. SHOWN ARE FOR MIRROR CELL MOUNTING
LONGITUDINAL SECTION - A-A
CROSS SECTION - B-B
.110
.230
12.9075°
.441
ENTRANCE
.125
FOCAL PLANE
232.94 mm
1.8328"
.04
.04
DETAIL - SERRATIONS, ALL FLAT SURFACES ADJACENT TO LIGHT PATH.
SCALE - 4:1

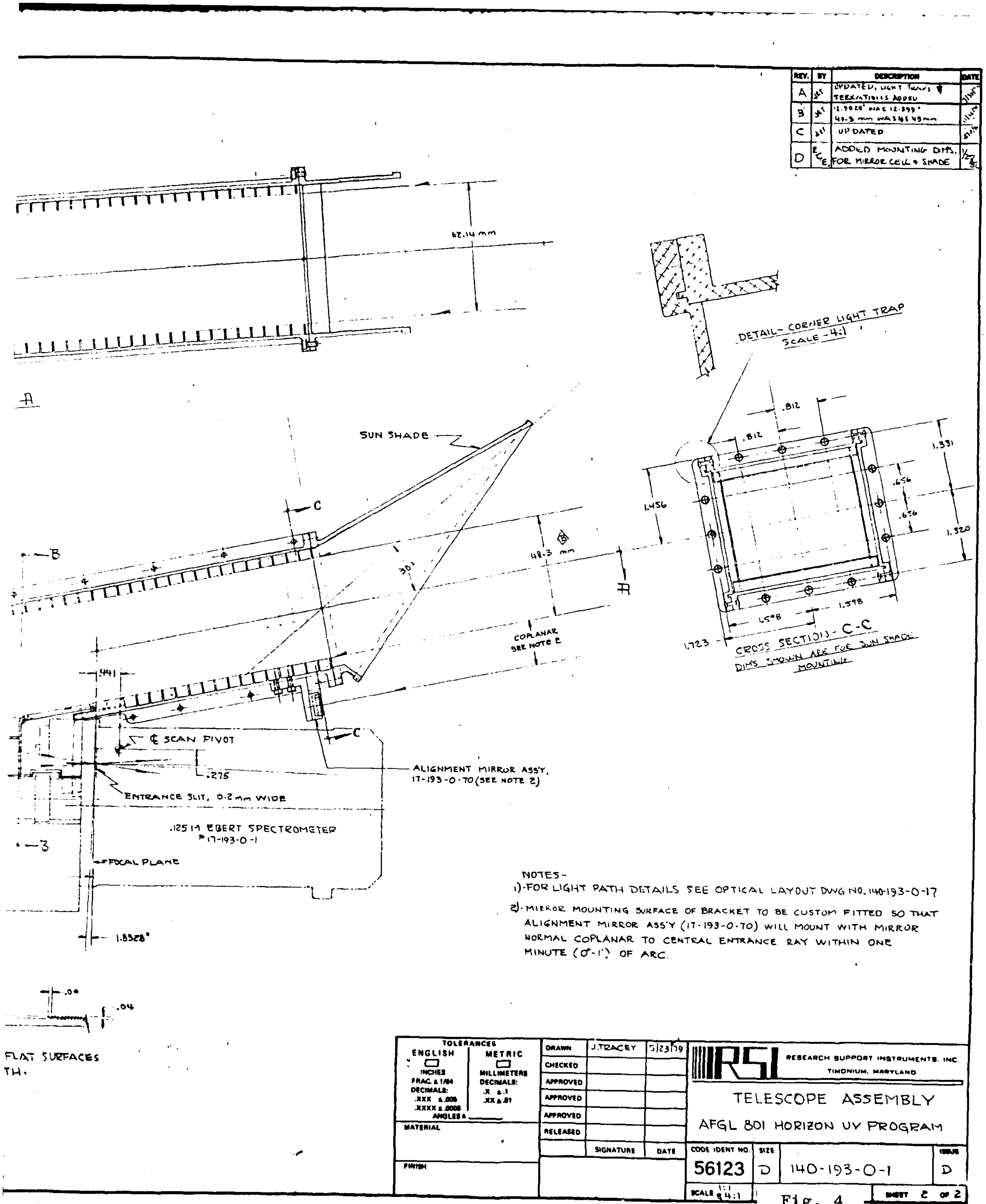
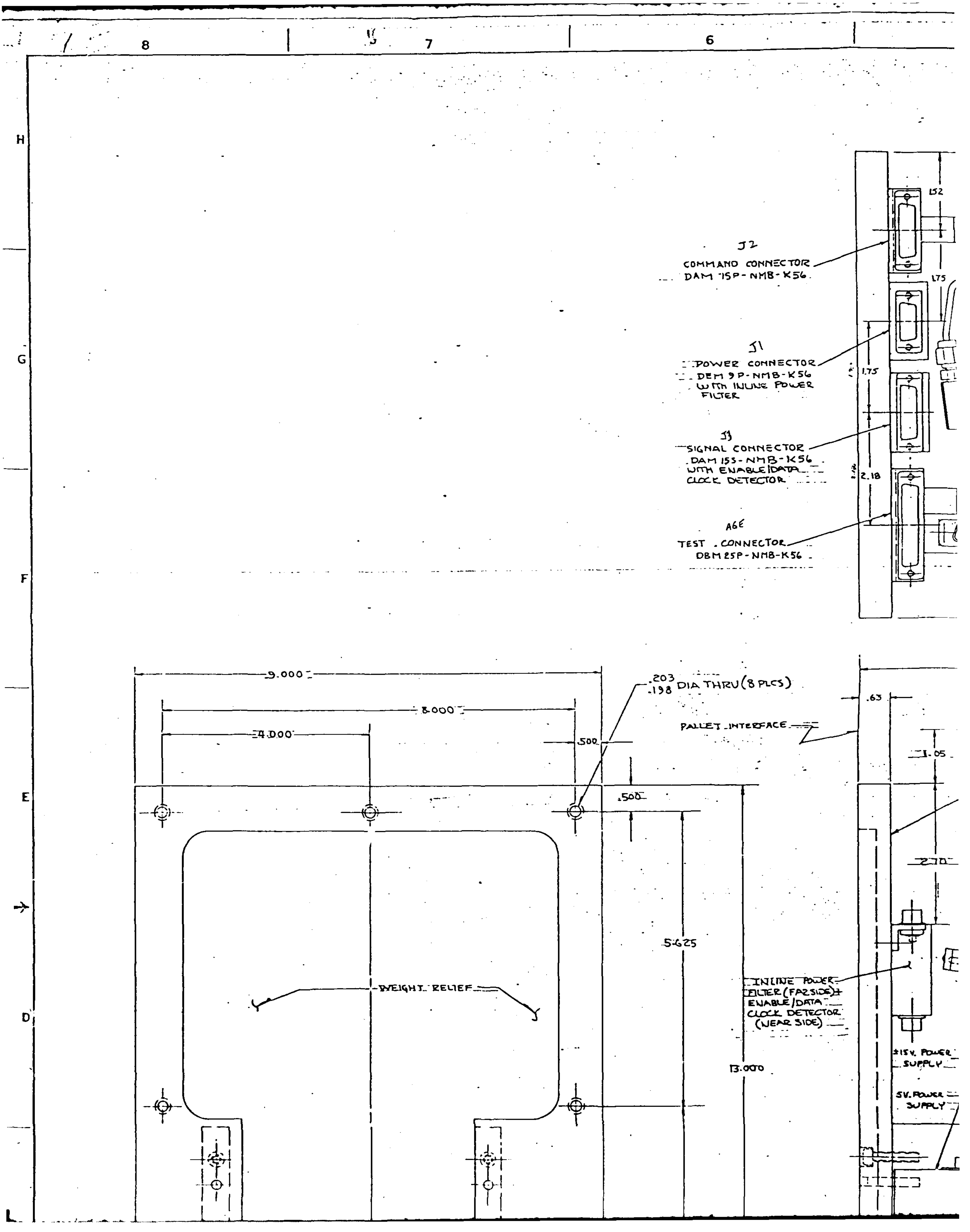

REV. | BY | DESCRIPTION | DATE
A UPDATED, LIGHT TRAPS & SERRATIONS ADDED
B "2.9028" WAS 12.899" 48.3 mm WAS 41.45mm
C UPDATED
D ADDED MOUNTING DIMS. FOR MIRROR CELL + SHADE
62.14 mm
DETAIL - CORNER LIGHT TRAP SCALE - 4:1
SUN SHADE
C
B
30°
48.3 mm
A
.812
1.331
.656
1.456
1.320
1.598
1.598
1.723
CROSS SECTION - C-C
DIMS SHOWN ARE FOR SUN SHADE MOUNTING
COPLANAR SEE NOTE 2
.441
℄ SCAN PIVOT
.275
ALIGNMENT MIRROR ASS'Y, 17-193-0-70 (SEE NOTE 2)
ENTRANCE SLIT, 0.2mm WIDE
.125 M EBERT SPECTROMETER #17-193-0-1
FOCAL PLANE
1.8328°
.04
.04
FLAT SURFACES
NOTES -
1) - FOR LIGHT PATH DETAILS SEE OPTICAL LAYOUT DWG NO. 140-193-0-17
2) - MIRROR MOUNTING SURFACE OF BRACKET TO BE CUSTOM FITTED SO THAT ALIGNMENT MIRROR ASS'Y (17-193-0-70) WILL MOUNT WITH MIRROR NORMAL COPLANAR TO CENTRAL ENTRANCE RAY WITHIN ONE MINUTE (0'-1') OF ARC.
TOLERANCES
ENGLISH INCHES FRAC. ± 1/64 DECIMALS: .XXX ± .005 .XXXX ± .0005 ANGLES ±
METRIC MILLIMETERS DECIMALS: .X ± .1 .XX ± .01
MATERIAL
FINISH
DRAWN J.TRACEY 5/23/79
CHECKED
APPROVED
APPROVED
APPROVED
RELEASED
SIGNATURE DATE
RSI RESEARCH SUPPORT INSTRUMENTS, INC. TIMONIUM, MARYLAND
TELESCOPE ASSEMBLY
AFGL 801 HORIZON UV PROGRAM
CODE IDENT NO. 56123
SIZE D
140-193-0-1
ISSUE D
SCALE 1:1 4:1
SHEET 2 OF 2

Fig. 4

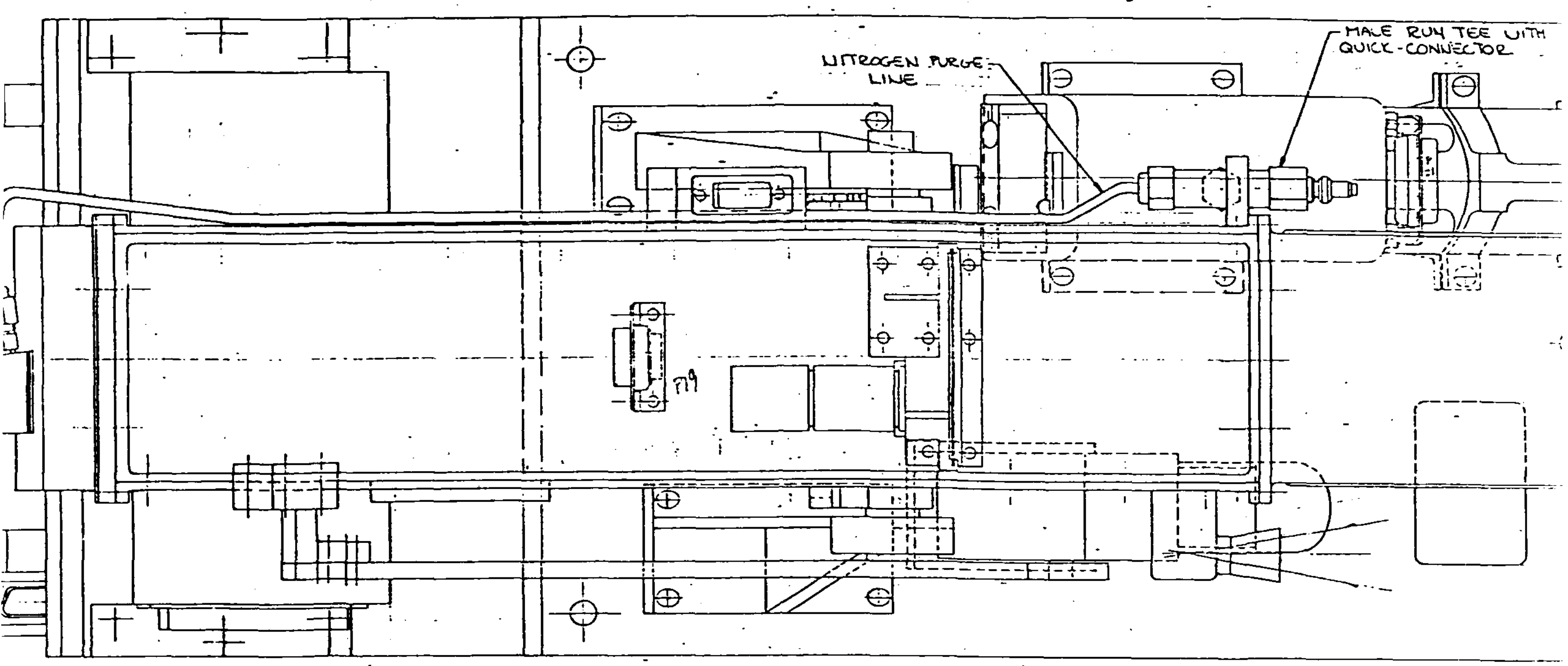


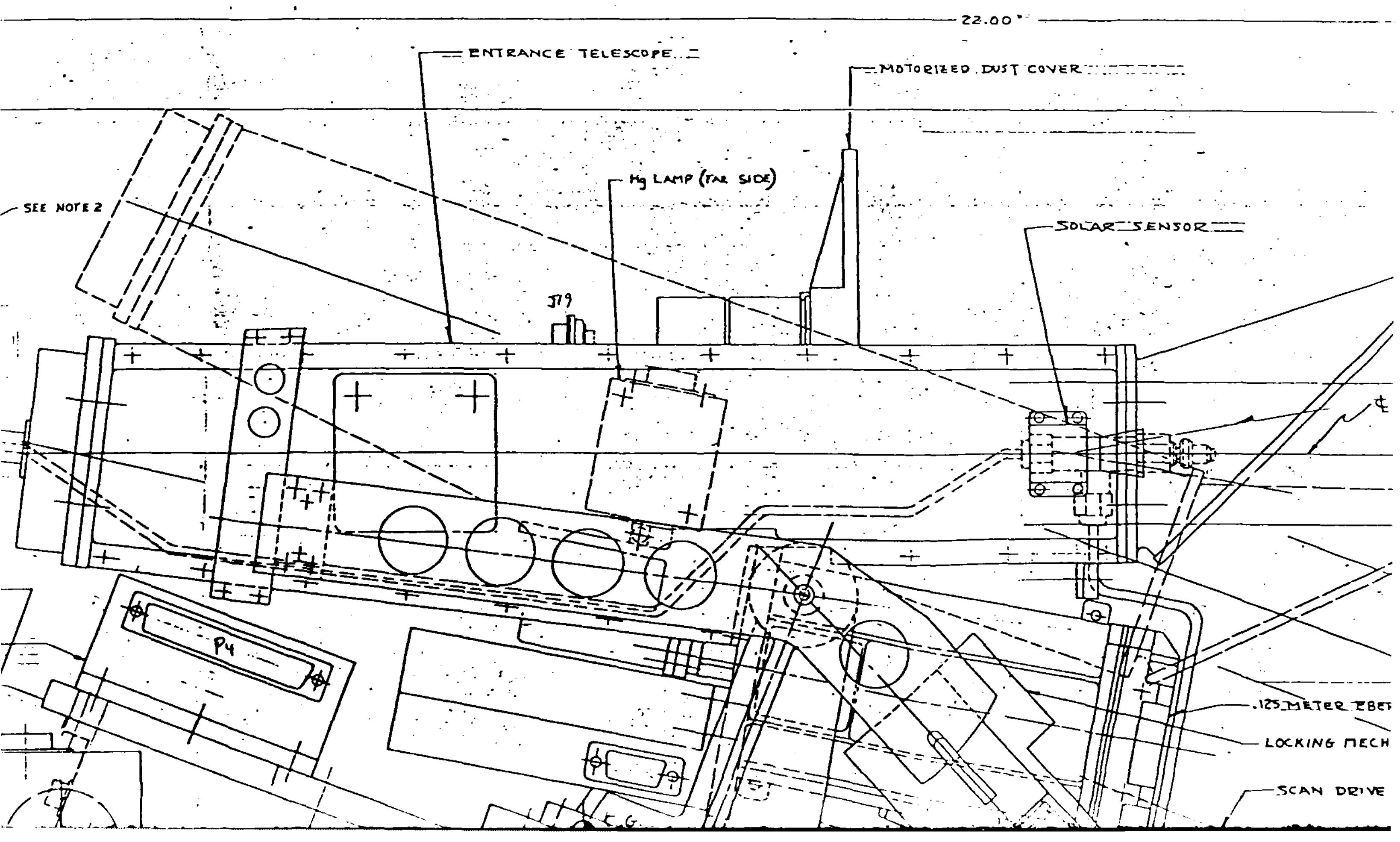
8
7
6
H
G
F
E
D
1.52
1.75
1.75
2.18
J2
COMMAND CONNECTOR
DAM 15P-NMB-K56
J1
POWER CONNECTOR
DEM 9P-NMB-K56
WITH INLINE POWER
FILTER
J3
SIGNAL CONNECTOR
DAM 15S-NMB-K56
WITH ENABLE/DATA
CLOCK DETECTOR
TEST CONNECTOR
DBM 25P-NMB-K56
9.000
8.000
4.000
.500
.500
.203
.198 DIA THRU (8 PLCS)
PALLET INTERFACE
.63
1.05
2.70
5.625
13.000
WEIGHT RELIEF
INLINE POWER
FILTER (FAR SIDE) +
ENABLE/DATA
CLOCK DETECTOR
(NEAR SIDE)
±15V. POWER
SUPPLY
5V. POWER
SUPPLY

5 4 3 2

SYM	ZONE	DESCRIPTION	DATE	APPROVED
H		UPDATED ELECTRONICS MODULES	10-16-81	R.EPPIG
I		MOVED C.G., 9.44 was 9.31, 4.62 was 4.50, 4.[illegible] was 4.[illegible]	10-21-81	R.EPPIG
J		ADDED INLINE POWER FILTER	11-11-81	R.EPPIG
K		ADDED NITROGEN PURGE LINE, + ENABLE/DATA CLOCK DETECTOR	1-10-84	R.EPPIG
L		ADDED OPTICS CUBE, SEE ECO 634	2-13-8[illegible]	R.C.E.

SYM	ZONE	DESCRIPTION	DATE	APPROVED
D		ADDED 25 PIN CONNECTOR, OPENED FIELD OF VIEW	7-9-81	R.EPPIG
E		TEST CONN. WAS A.G.E. CONN, ADDED 'N' PURGE, EXHAUST PORT, + Hg LAMP	8-4-81	R.EPPIG
F		ADDED CENTER OF GRAVITY	9-18-81	R.EPPIG
G		UPDATED ELECTRONICS MODULES, 5.0lbs was 4.0lbs ADDED CUT OUT	10-6-81	R.EPPIG



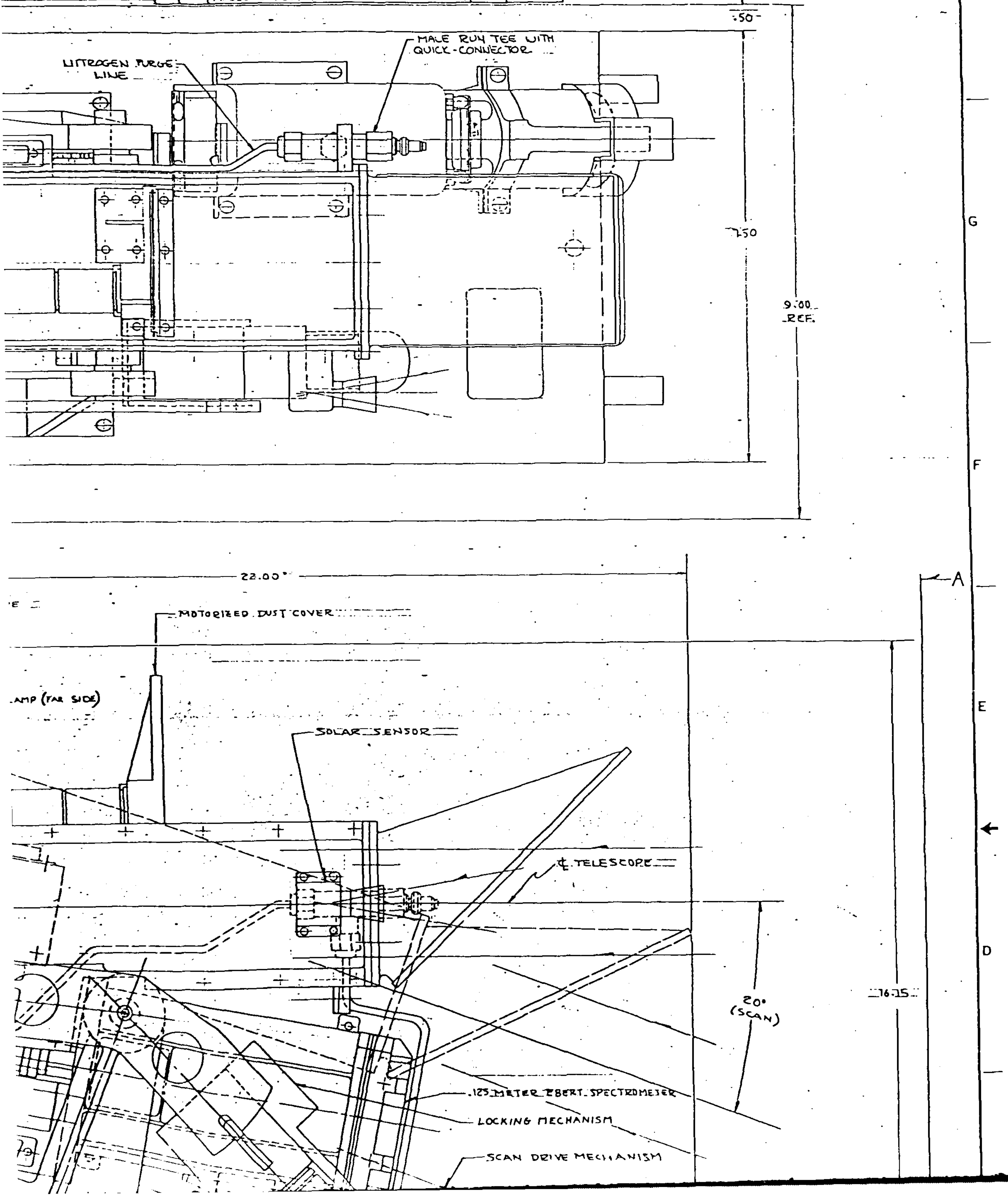


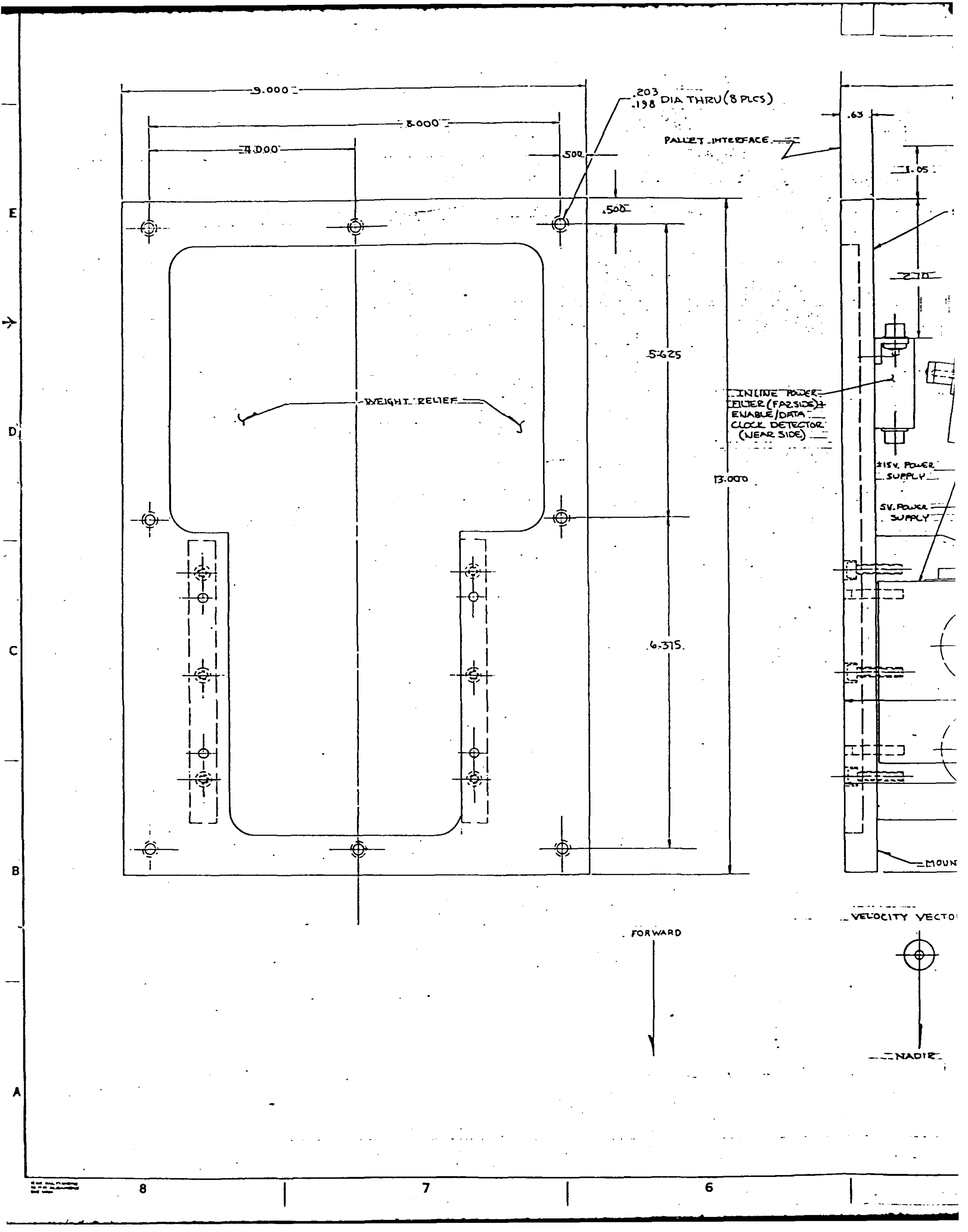
DESCRIPTION	DATE	APPROVED
	10-16-81	R.EPPIG
,... WAS 4.?	10-21-81	R.EPPIG
	11-11-81	R.EPPIG
.../DATA	1-10-84	R.EPPIG
	2-13-8?	R.C.E.

SYM	ZONE	DESCRIPTION	DATE	APPROVED
D	---	ADDED 25 PIN CONNECTOR, OPENED FIELD OF VIEW	7-9-81	R.EPPIG
E	-	TEST CONN. WAS A.G.E. CONN, ADDED N PURGE, EXHAUST PORT, + Hg LAMP	8-4-81	R.EPPIG
F		ADDED CENTER OF GRAVITY	9-18-81	R.EPPIG
G		UPDATED ELECTRONICS MODULES, 5.0LBS WAS 4.0LB ADDED CUT OUT	10-6-81	R.EPPIG

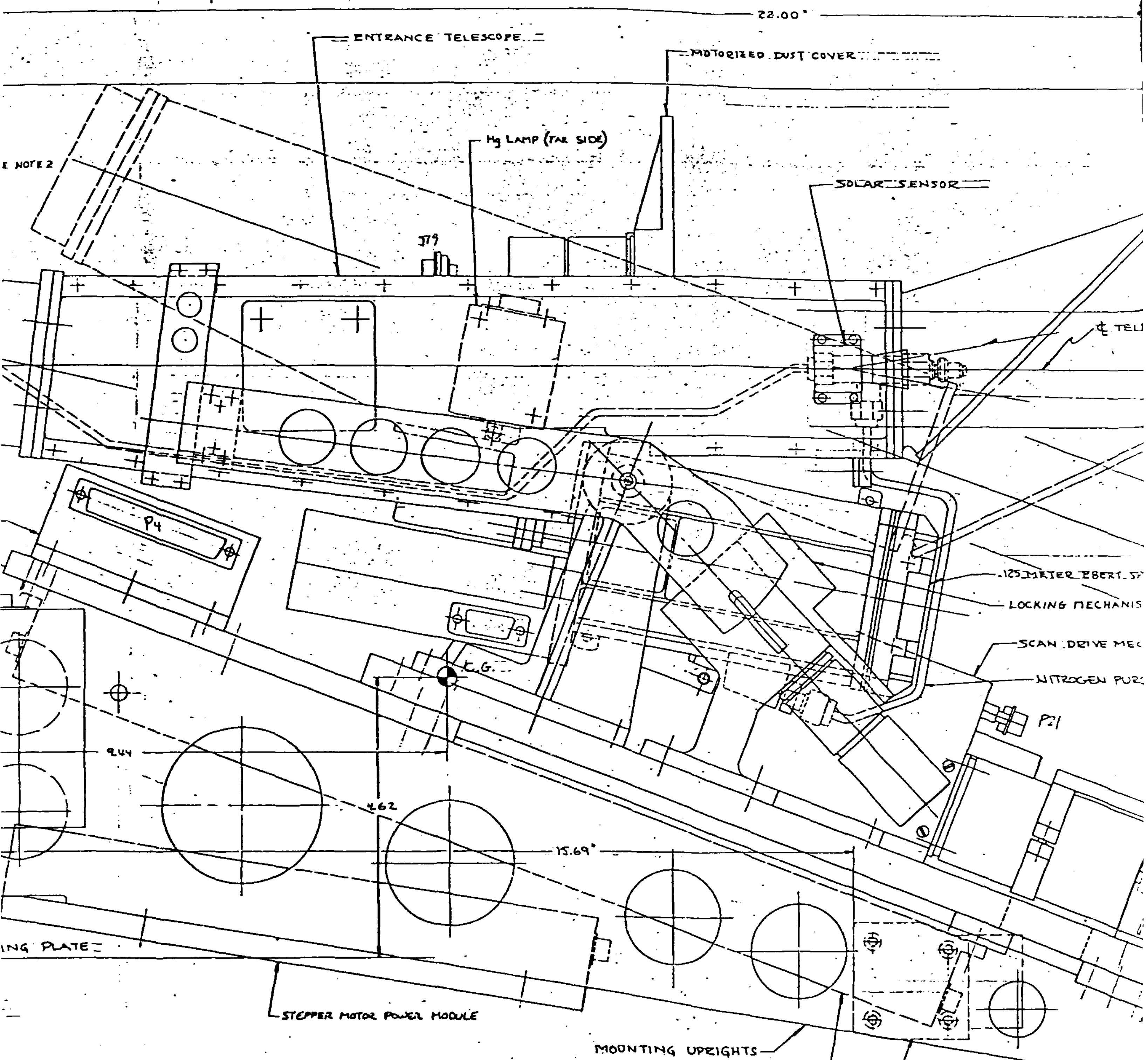
REVISIONS

SYM	ZONE	DESCRIPTION	DATE	APPROVED
A		REDRAWN & UPDATED	6/11/81	
B		4-193-0-22 WAS 3-193-0-2, LOCK ADDED, + UPDATED	6-18-81	R.EPPIG
C		UPDATED, 5.0lb WAS 3.0, 11.0lb WAS 14.0	6-29-81	R.EPPIG





9.000
8.000
4.000
.502
.203
.198 DIA THRU (8 PLCS)
PALLET INTERFACE
.63
1.05
2.70
.500
5.625
6.315
13.000
WEIGHT RELIEF
INLINE POWER FILTER (FAR SIDE)
ENABLE/DATA CLOCK DETECTOR (NEAR SIDE)
±15V. POWER SUPPLY
5V. POWER SUPPLY
MOUN
FORWARD
VELOCITY VECTO
NADIR
E
D
C
B
A
8
7
6



NOTES

1) APPROXIMATE WEIGHTS
   a) SPECTROMETER & TELESCOPE - 5.0 lb.
   b) SCAN PLATFORM & MECHANISM - 5.0 lb. ©
   c) MOUNTING PLATE & UPRIGHTS - 11.0 lb. ©
   d) ELECTRONICS MODULES - 5.0 lb ©
   e) CABLING/HARNESS - 2.5 lb.

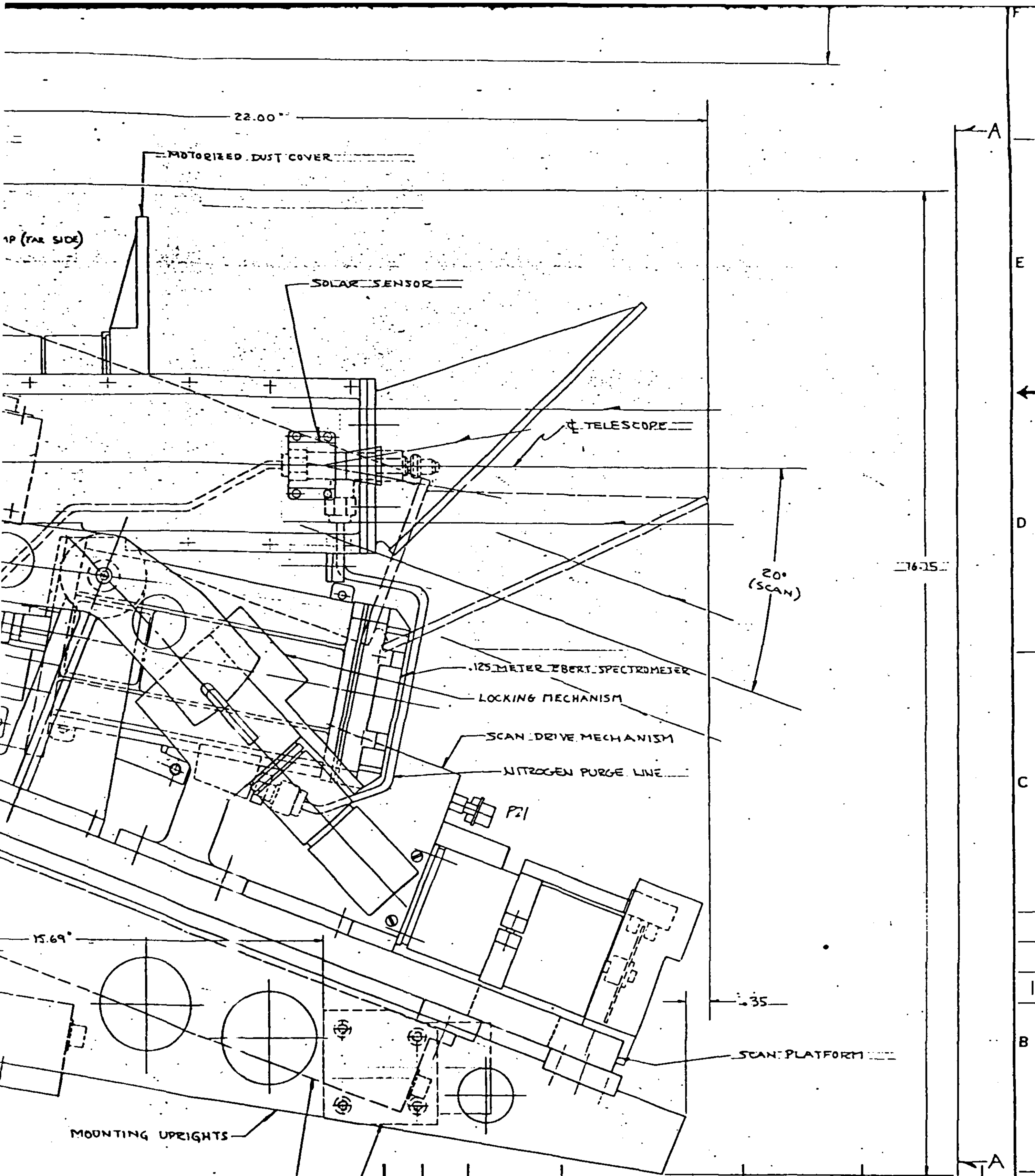
2) FORWARD DIRECTION TO BE ENGRAVED ON BASE PLATE IN APPROX. POSITION SHOWN

QTY. REQ'D.	CODE IDENT.	PART NO. OR IDENTIFYING NO.

TOLERANCES		
ENGLISH INCHES	METRIC MILLIMETERS	DRAWN R.E
FRAC. ± 1/64	DECIMALS:	CHECKED
DECIMALS: .XXX ± .005 .XXXX ± .0005	.X ± .1 .XX ± .01	APPROVED
ANGLES ±		APPROVED
		APPROVED
MATERIAL		RELEASED
FINISH		SIG

NEXT ASSEMBLY	USED ON	NEXT ASSY	FINAL ASSY
APPLICATION		QUANTITY	

5 4 3 2



QTY. REQ'D.	CODE IDENT.	PART NO. OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION	MATERIAL DESCRIPTION	MATERIAL SPECIFICATION	ITEM NO.

LIST OF MATERIAL

TOLERANCES
ENGLISH: INCHES; FRAC. ± 1/64; DECIMALS: .XXX ± .005, .XXXX ± .0005; ANGLES ±
METRIC: MILLIMETERS; DECIMALS: .X ± .1, .XX ± .01

MATERIAL

FINISH

DRAWN	R. EPPIG	4/11/81
CHECKED	J. Gray	
APPROVED		
APPROVED		
APPROVED		
RELEASED		
	SIGNATURE	DATE

NEXT ASSEMBLY	USED ON	NEXT ASSY.	FINAL ASSY.
APPLICATION		QUANTITY	

RSI RESEARCH SUPPORT INSTRUMENTS, INC.
TIMONIUM, MARYLAND

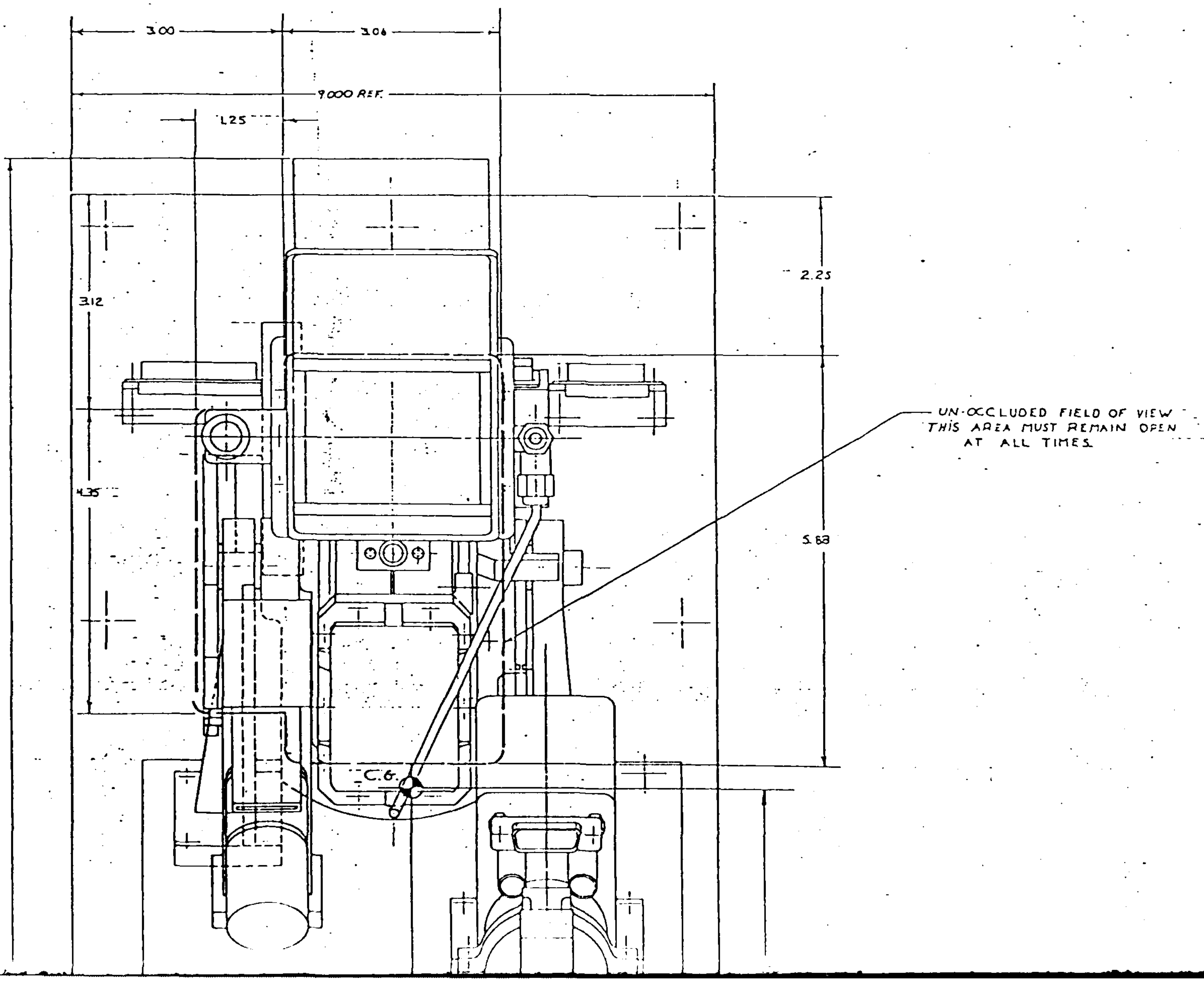
INTERFACE DRAWING
SCAN PLATFORM TO SHUTTLE PALLET
AFGL 801A SHUTTLE SETS-1 PALLET

CODE IDENT NO.	SIZE		ISSUE
56123	E	4-193-0-22	L
SCALE 1:1		SHEET 1 OF 2	

Fig. 5

8 7 6

H
G
F
E
D


3.00
3.06
9.000 REF.
1.25
3.12
4.35
2.25
5.63
C.G.
UN-OCCLUDED FIELD OF VIEW
THIS AREA MUST REMAIN OPEN
AT ALL TIMES

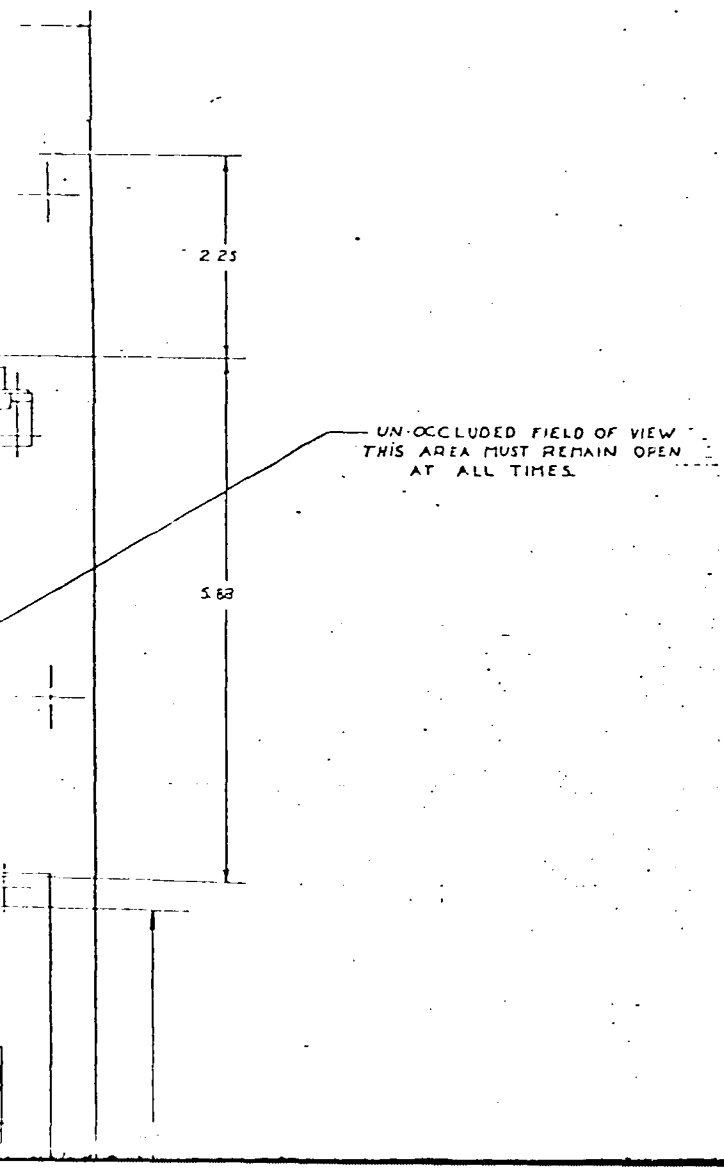
3 2 1

REVISIONS				
SYM	ZONE	DESCRIPTION	DATE	APPROVED

H

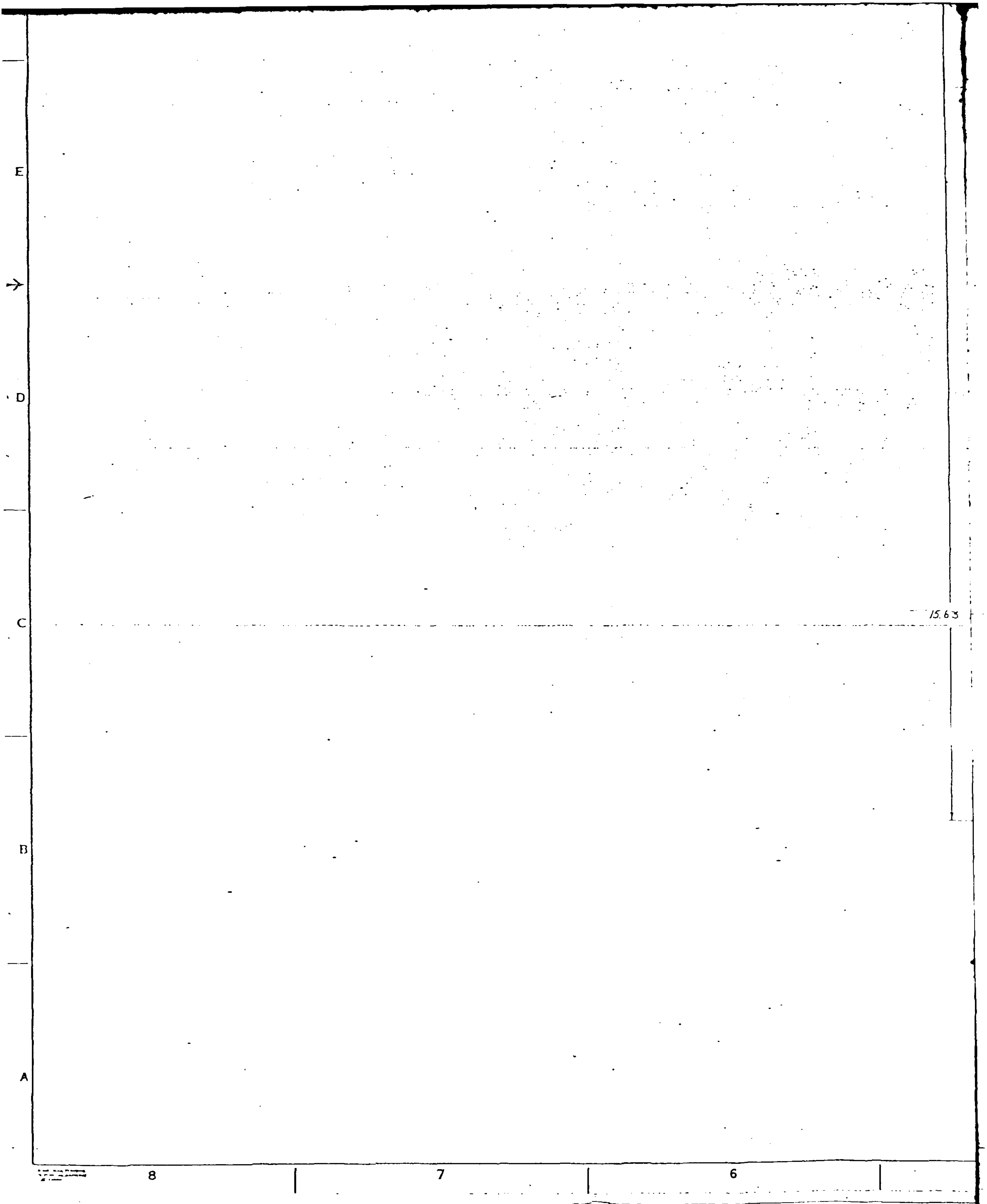
G

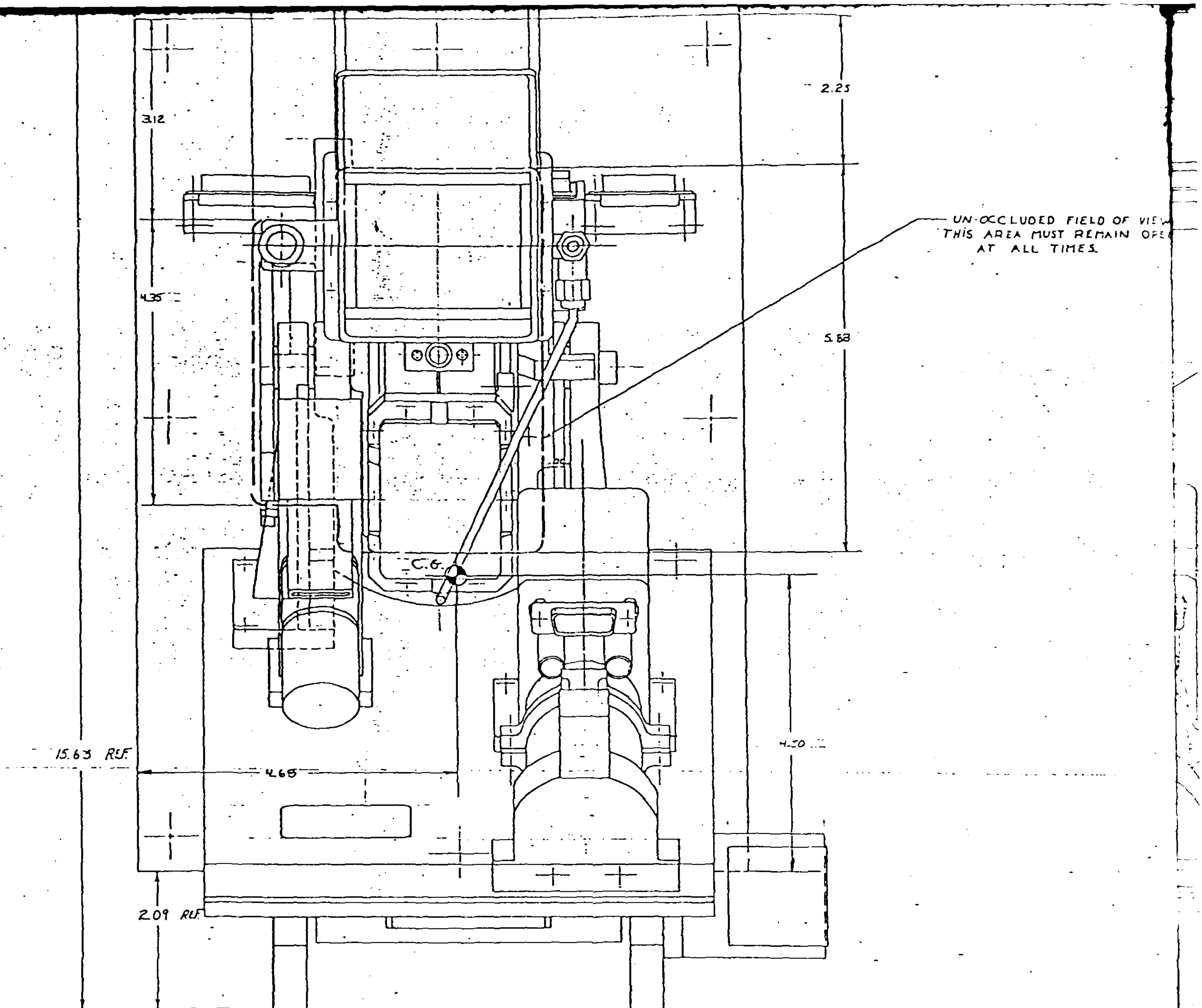
F



E

D


E
D
C
B
A
8
7
6
15.63



VIEW A-A
FROM SHEET 1

NEXT ASSEMBLY	USED ON	NEXT ASSY	FINAL ASSY	
APPLICATION		QUANTITY		

QTY. REQ'D. | CODE IDENT

TOLERA
ENGLISH
INCHES
FRAC. ± 1/64
DECIMALS:
.XXX ± .005
.XXXX ± .0005
ANGLES ±

MATERIAL

FINISH

5 4 3

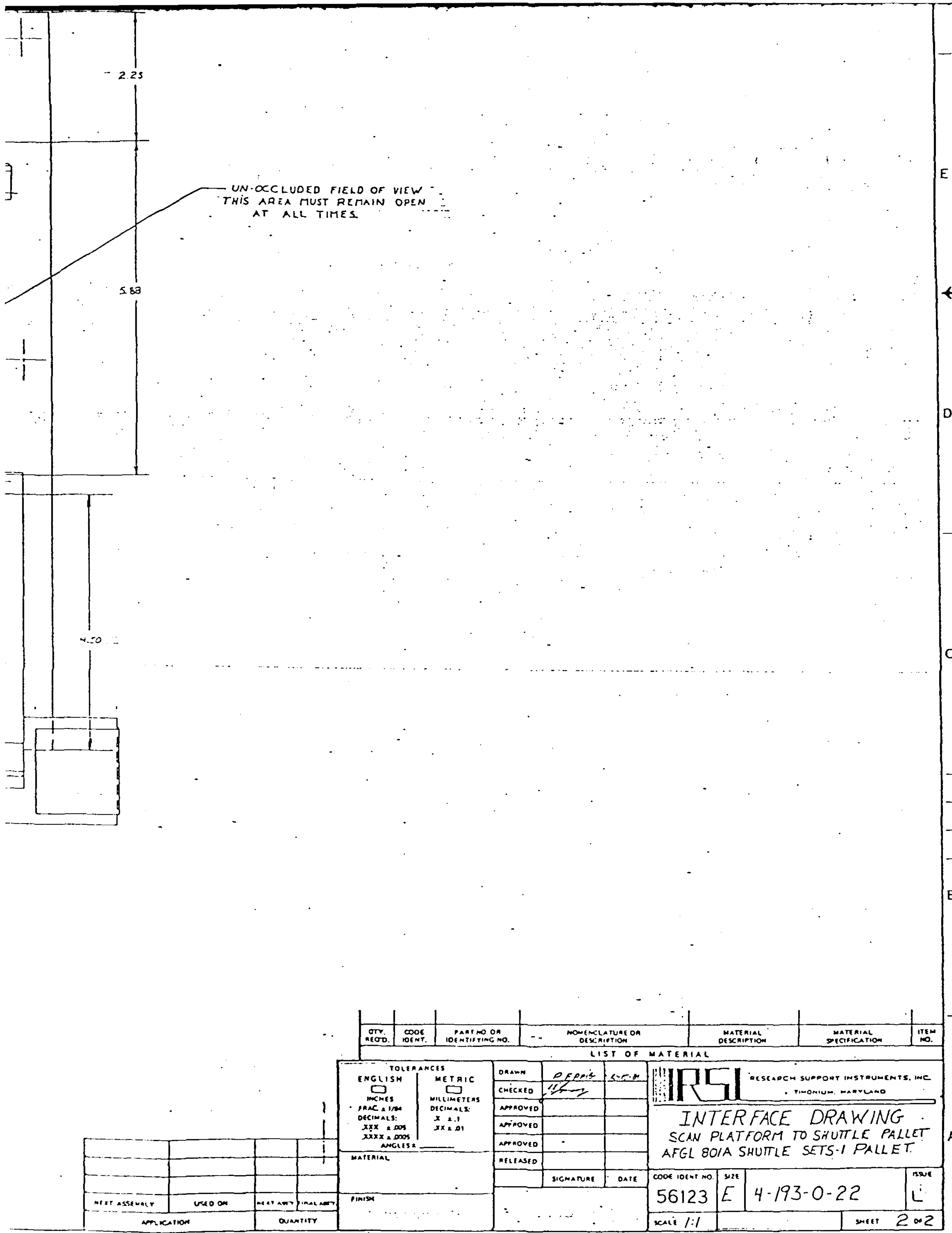
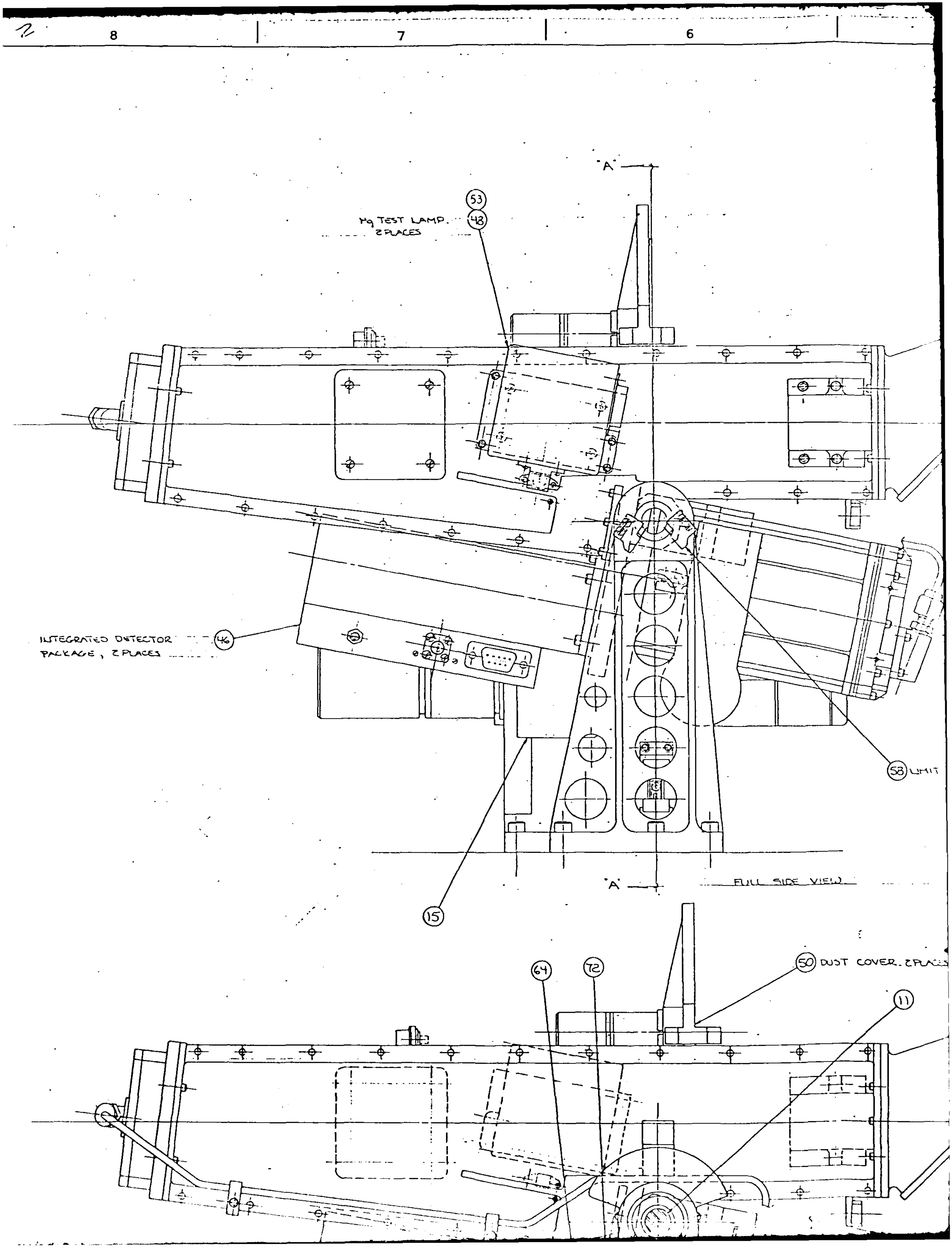
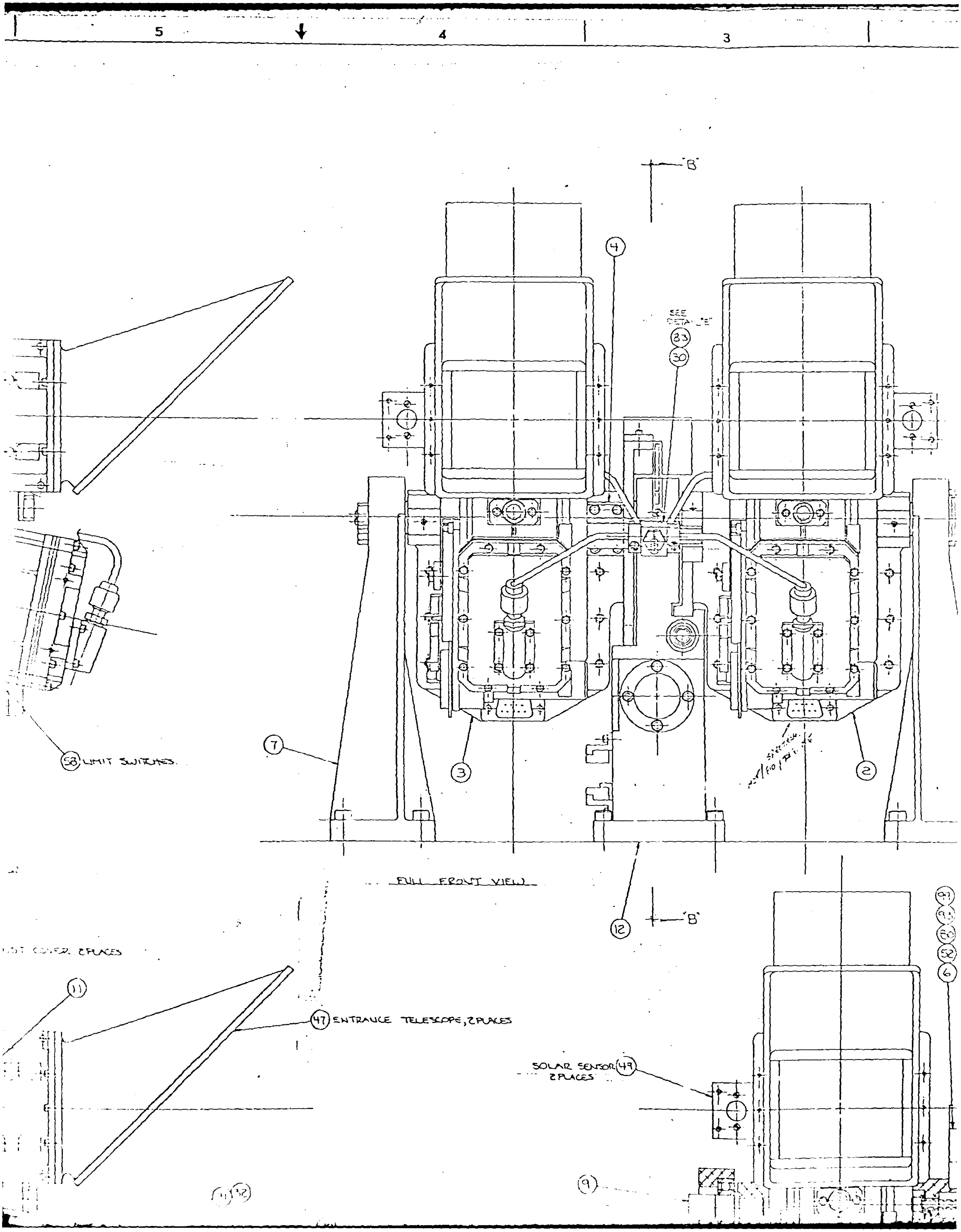

2.25
UN-OCCLUDED FIELD OF VIEW
THIS AREA MUST REMAIN OPEN
AT ALL TIMES
5.83
4.50
E
D
C
E
QTY. REQ'D.
CODE IDENT.
PART NO. OR IDENTIFYING NO.
NOMENCLATURE OR DESCRIPTION
MATERIAL DESCRIPTION
MATERIAL SPECIFICATION
ITEM NO.
LIST OF MATERIAL
TOLERANCES
ENGLISH
INCHES
FRAC ± 1/64
DECIMALS:
.XXX ± .005
.XXXX ± .0005
ANGLES ±
METRIC
MILLIMETERS
DECIMALS:
.X ± .1
.XX ± .01
MATERIAL
FINISH
DRAWN
CHECKED
APPROVED
APPROVED
APPROVED
RELEASED
SIGNATURE
DATE
RSI
RESEARCH SUPPORT INSTRUMENTS, INC.
TIMONIUM, MARYLAND
INTERFACE DRAWING
SCAN PLATFORM TO SHUTTLE PALLET
AFGL 801A SHUTTLE SETS-1 PALLET
CODE IDENT NO.
56123
SIZE
E
4-193-0-22
ISSUE
L
SCALE 1:1
SHEET 2 OF 2
NEXT ASSEMBLY
USED ON
NEXT ASSY
FINAL ASSY
APPLICATION
QUANTITY
3
2
1

Fig. 5



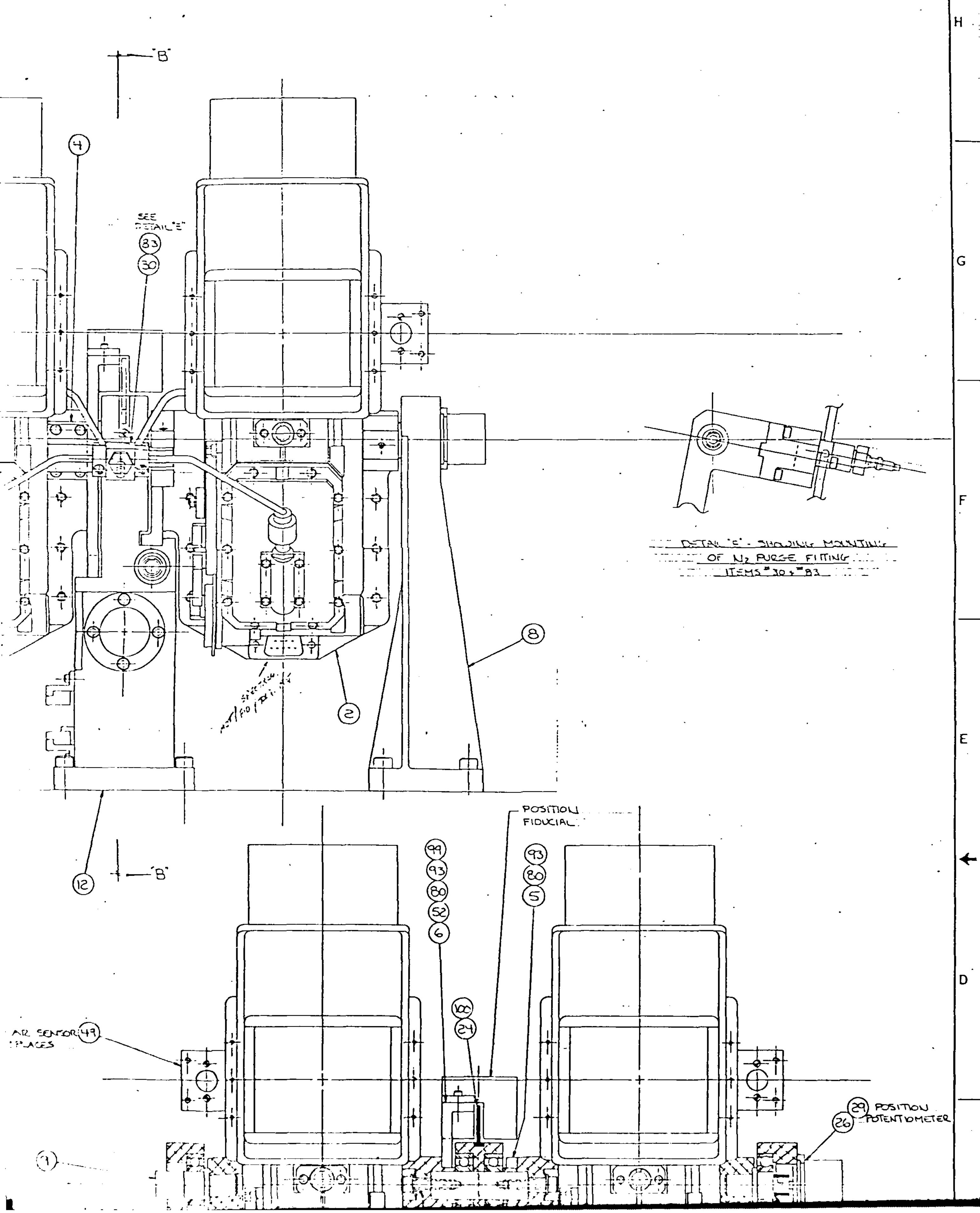
8
7
6
Hg TEST LAMP 2 PLACES
53
48
"A"
INTEGRATED DETECTOR PACKAGE, 2 PLACES
46
58 LIMIT
FULL SIDE VIEW
15
50 DUST COVER, 2 PLACES
64
72
11

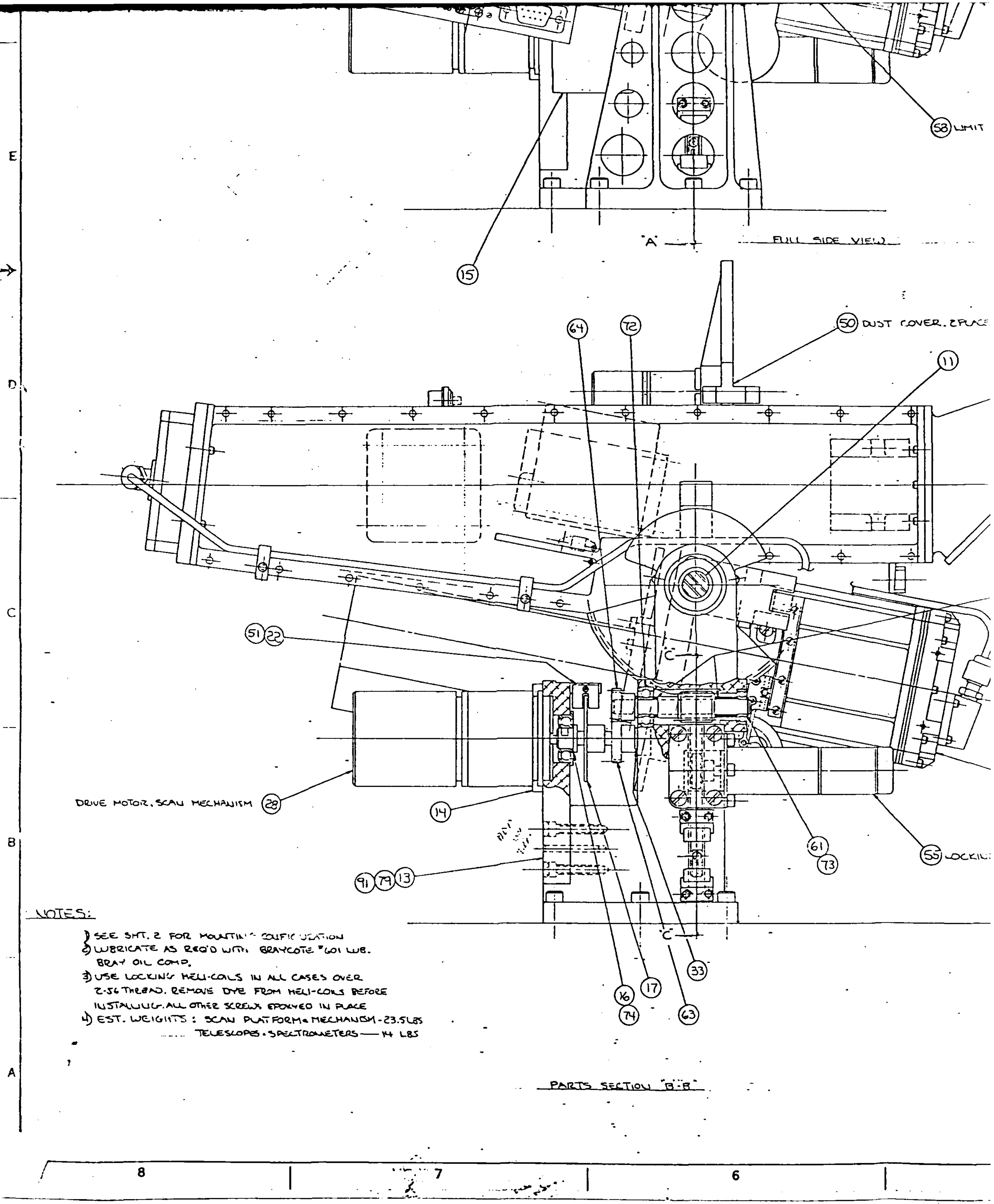


5
4
3
B
SEE DETAIL "E"
33
30
4
58 LIMIT SWITCHES
7
3
2
12
FULL FRONT VIEW
B
11
47 ENTRANCE TELESCOPE, 2 PLACES
SOLAR SENSOR 49
2 PLACES
6
52
9

3 2 1

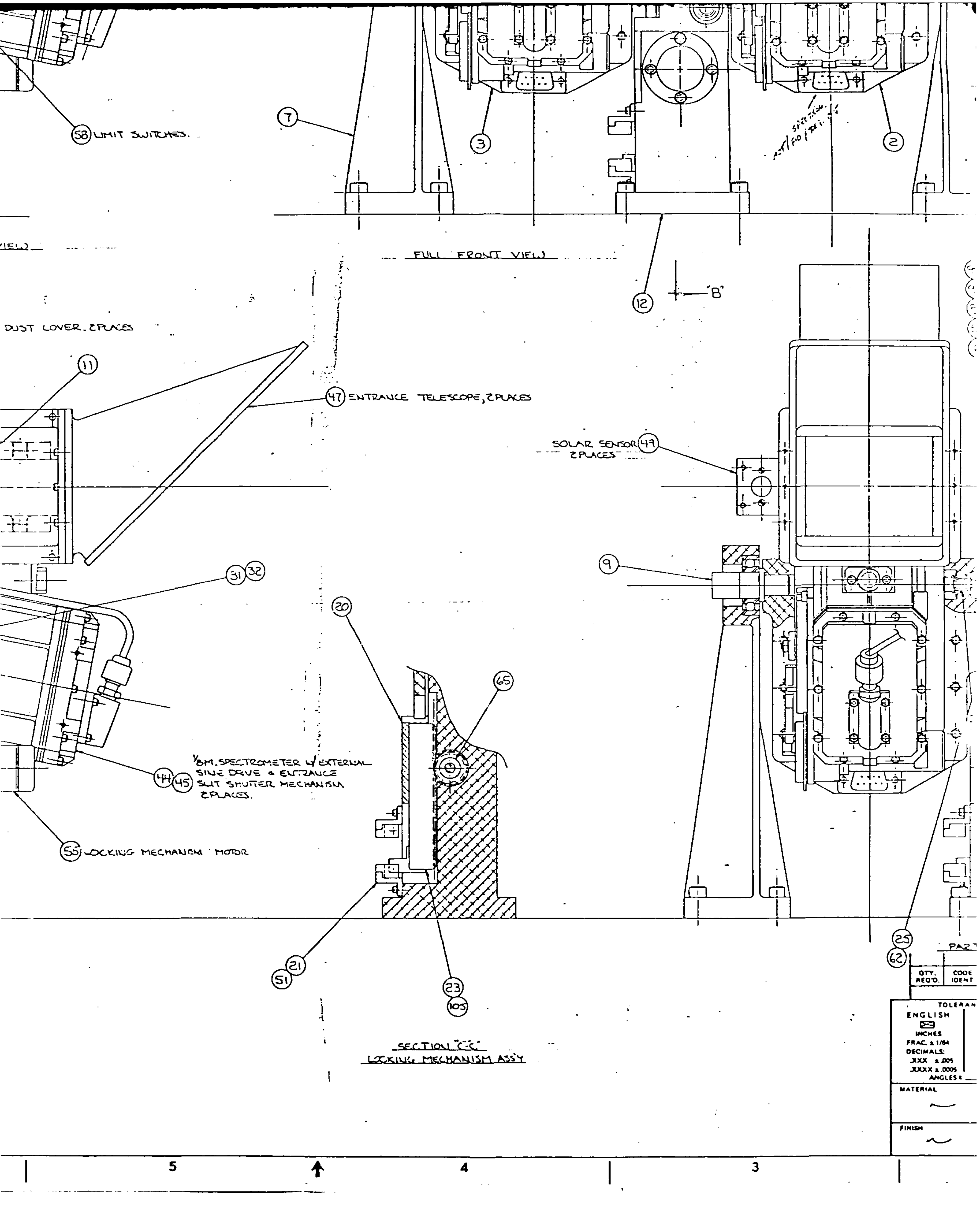
REVISIONS				
SYM	ZONE	DESCRIPTION	DATE	APPROVED
B		SEE ECO 530	6-1-84	R.C.E.
C		REVISED PER ECO 1198	6-19-87	M.YAKER



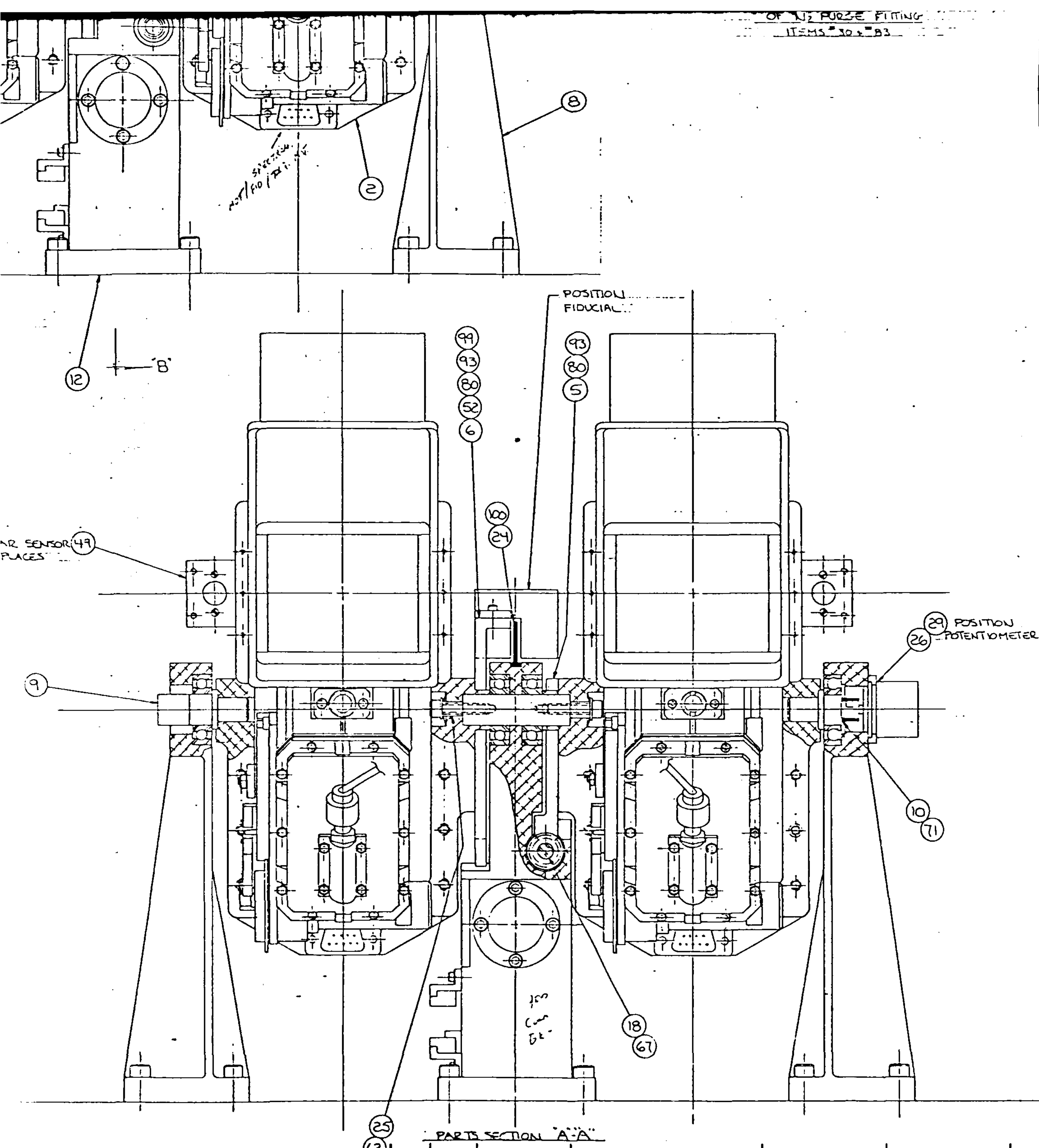


NOTES:

1) SEE SHT. 2 FOR MOUNTING CONFIGURATION
2) LUBRICATE AS REQ'D WITH BRAYCOTE #601 LUB. BRAY OIL COMP.
3) USE LOCKING HELI-COILS IN ALL CASES OVER 2-56 THREAD. REMOVE DYE FROM HELI-COILS BEFORE INSTALLING. ALL OTHER SCREWS EPOXYED IN PLACE
4) EST. WEIGHTS: SCAN PLATFORM & MECHANISM - 23.5 LBS
TELESCOPES & SPECTROMETERS — 14 LBS



58 LIMIT SWITCHES.
7
3
2
FULL FRONT VIEW
12
'B'
DUST COVER. 2 PLACES
11
47 ENTRANCE TELESCOPE, 2 PLACES
SOLAR SENSOR 49
2 PLACES
9
31 32
20
65
1/8 M. SPECTROMETER W/ EXTERNAL
SINE DRIVE & ENTRANCE
SLIT SHUTTER MECHANISM
2 PLACES.
44 45
55 LOCKING MECHANISM MOTOR
21
51
23
105
SECTION 'C-C'
LOCKING MECHANISM ASS'Y
25
62
PART
QTY. REQ'D.
CODE IDENT
TOLERAN
ENGLISH
INCHES
FRAC. ± 1/64
DECIMALS:
.XXX ± .005
.XXXX ± .0005
ANGLES ±
MATERIAL
FINISH
5
4
3



QTY. REQ'D.	CODE IDENT.	PART NO. OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION	MATERIAL DESCRIPTION	MATERIAL SPECIFICATION	ITEM NO.

LIST OF MATERIAL

TOLERANCES
ENGLISH: INCHES, FRAC. ± 1/64, DECIMALS: .XXX ± .005, .XXXX ± .0005, ANGLES ±
METRIC: MILLIMETERS, DECIMALS: .X ± .1, .XX ± .01

MATERIAL ~
FINISH ~

	SIGNATURE	DATE
DRAWN	REPPIG	6-1-84
CHECKED		
APPROVED		
APPROVED		
APPROVED		
RELEASED		

RSI RESEARCH SUPPORT INSTRUMENTS INC
COCKEYSVILLE MARYLAND

DUAL SCANNING SPECTROMETER ASSEMBLY

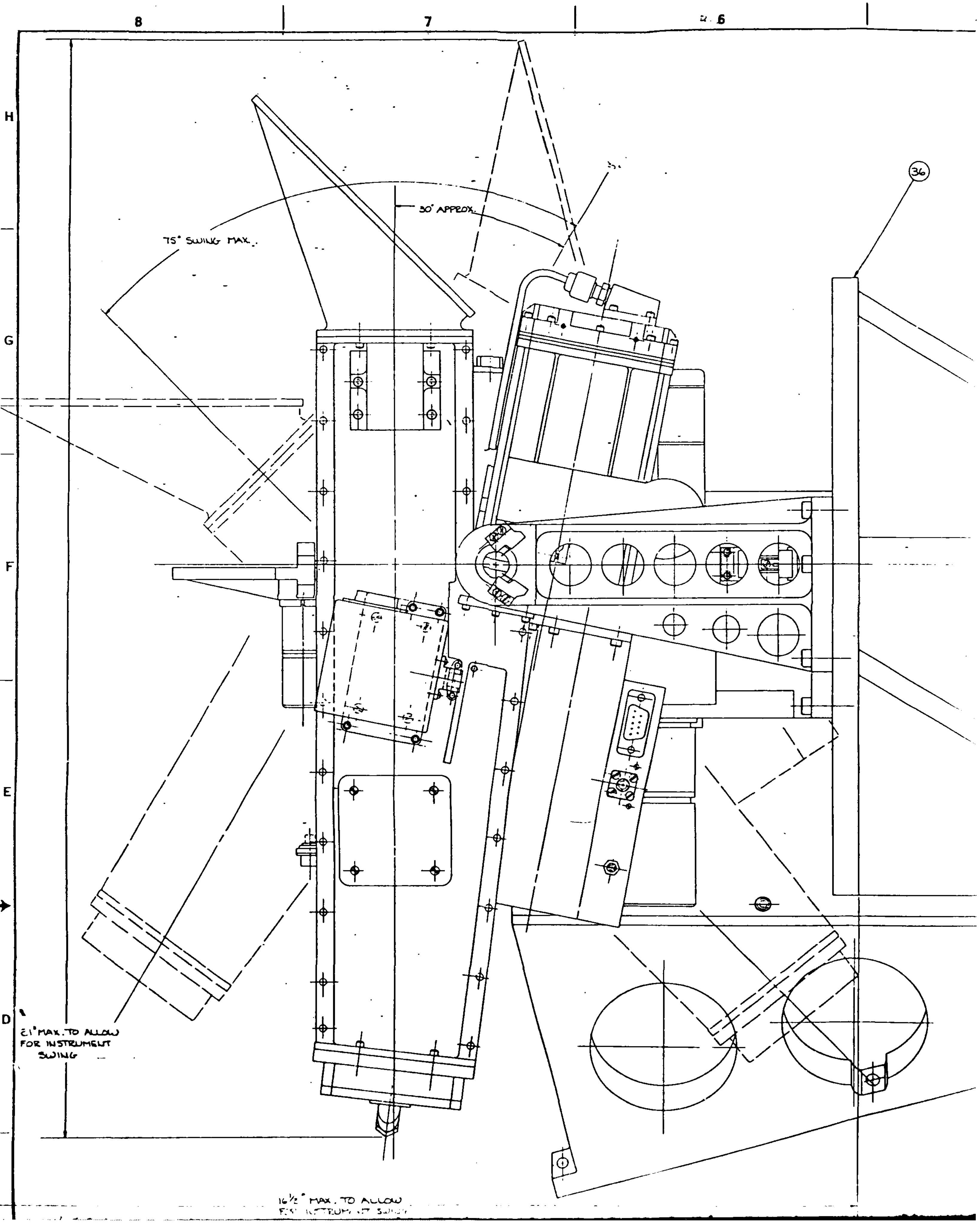
AFGL HUP PROGRAM, DR HUFFMAN

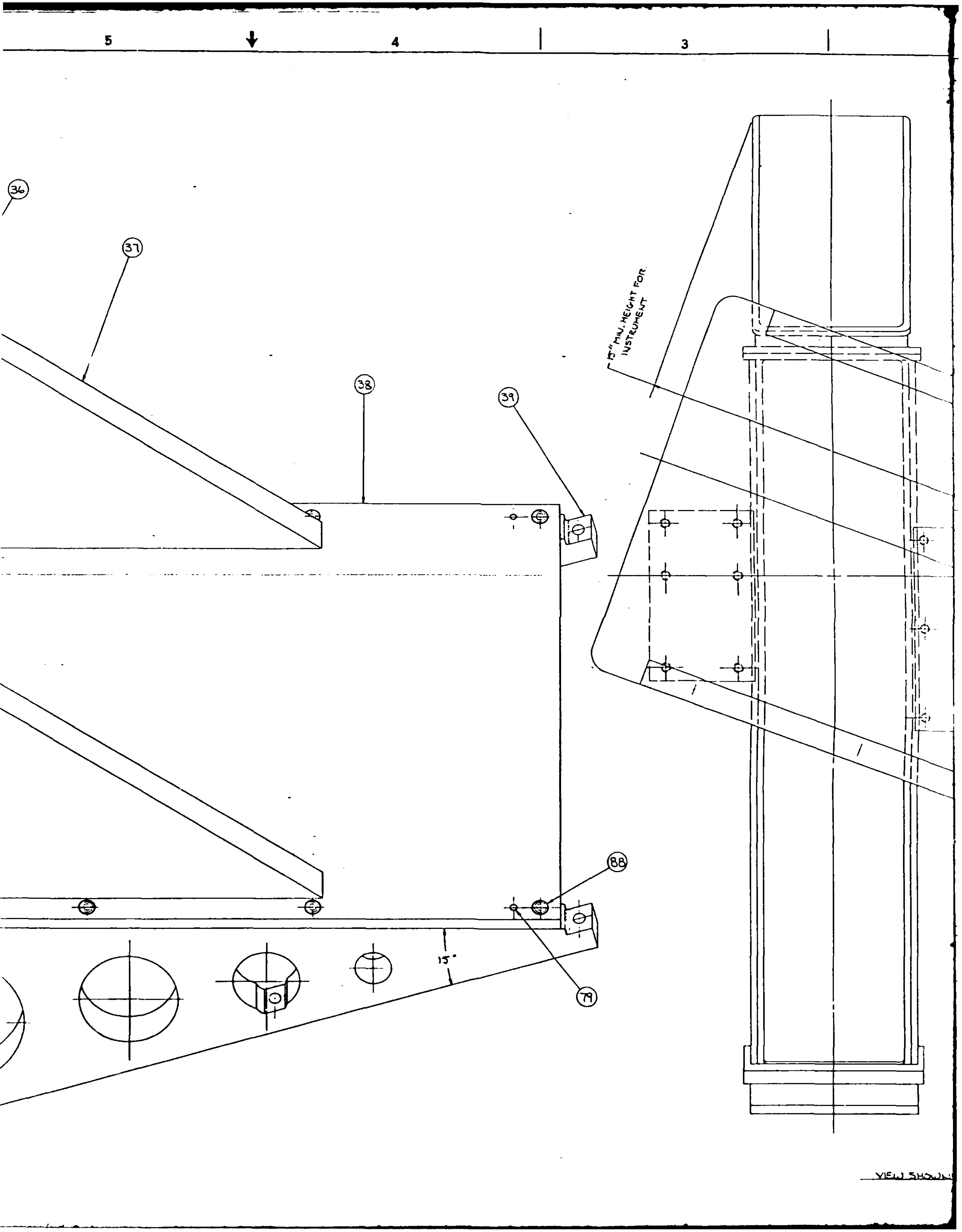
CODE IDENT NO.	SIZE		ISSUE
56123	E	0-224-0-1	C

SCALE 1:1 SHEET 1 OF 2

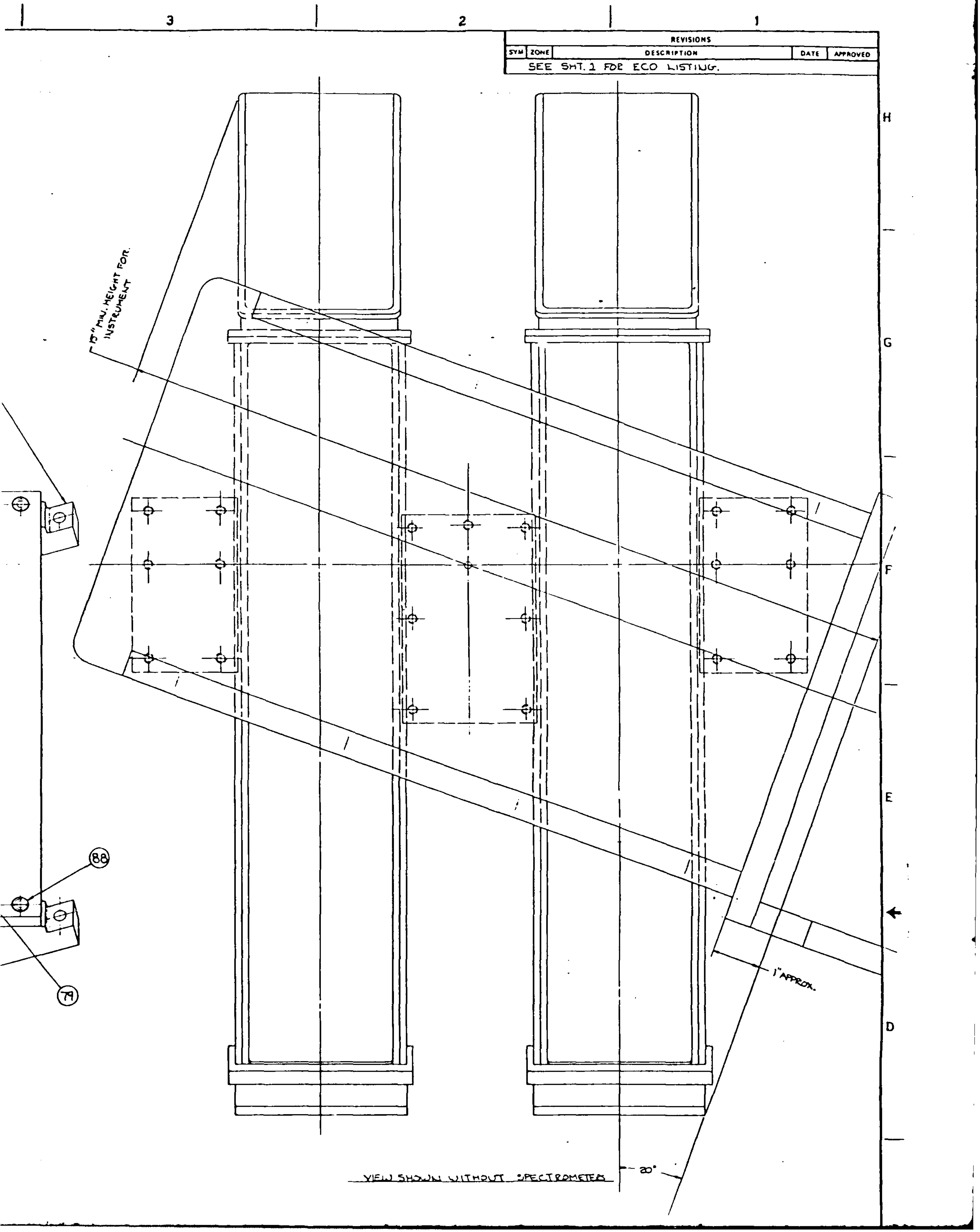
3 2 1

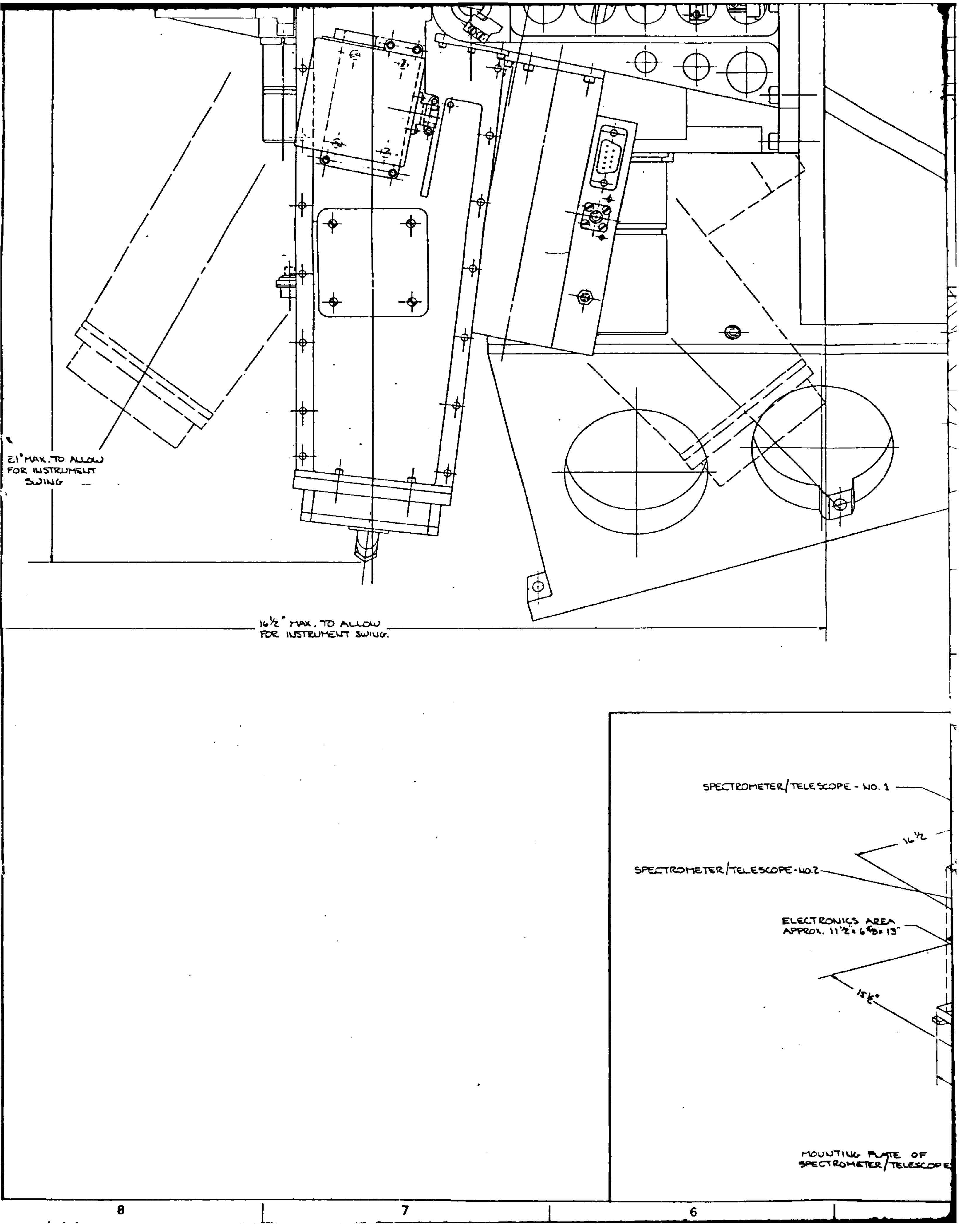
Fig. 6


8
7
6
H
G
F
E
D
75° SWING MAX.
30° APPROX.
36
21° MAX. TO ALLOW FOR INSTRUMENT SWING
16½° MAX. TO ALLOW

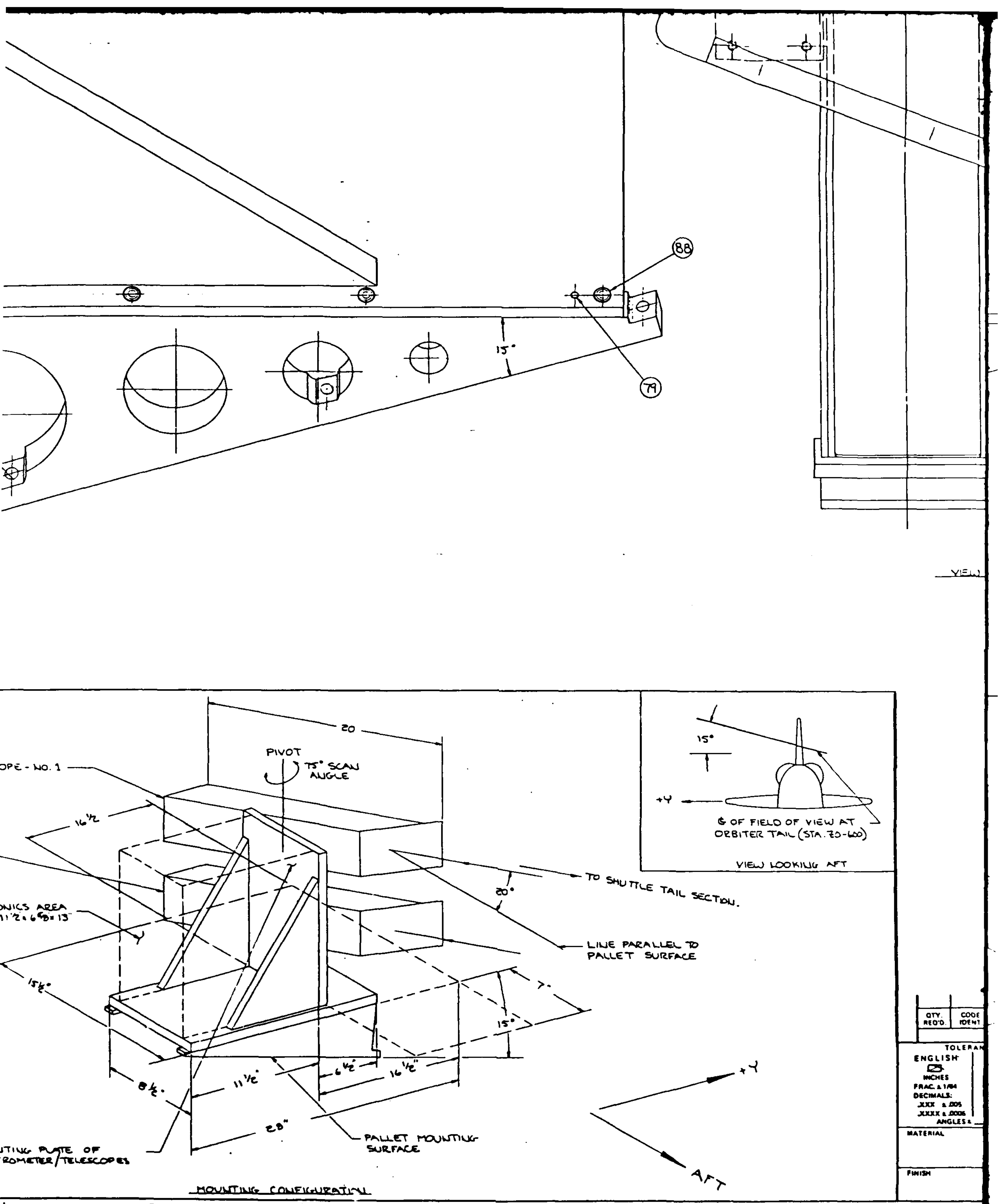

5
4
3
36
37
38
39
15" MIN. HEIGHT FOR INSTRUMENT
88
79
15°
VIEW SHOWN

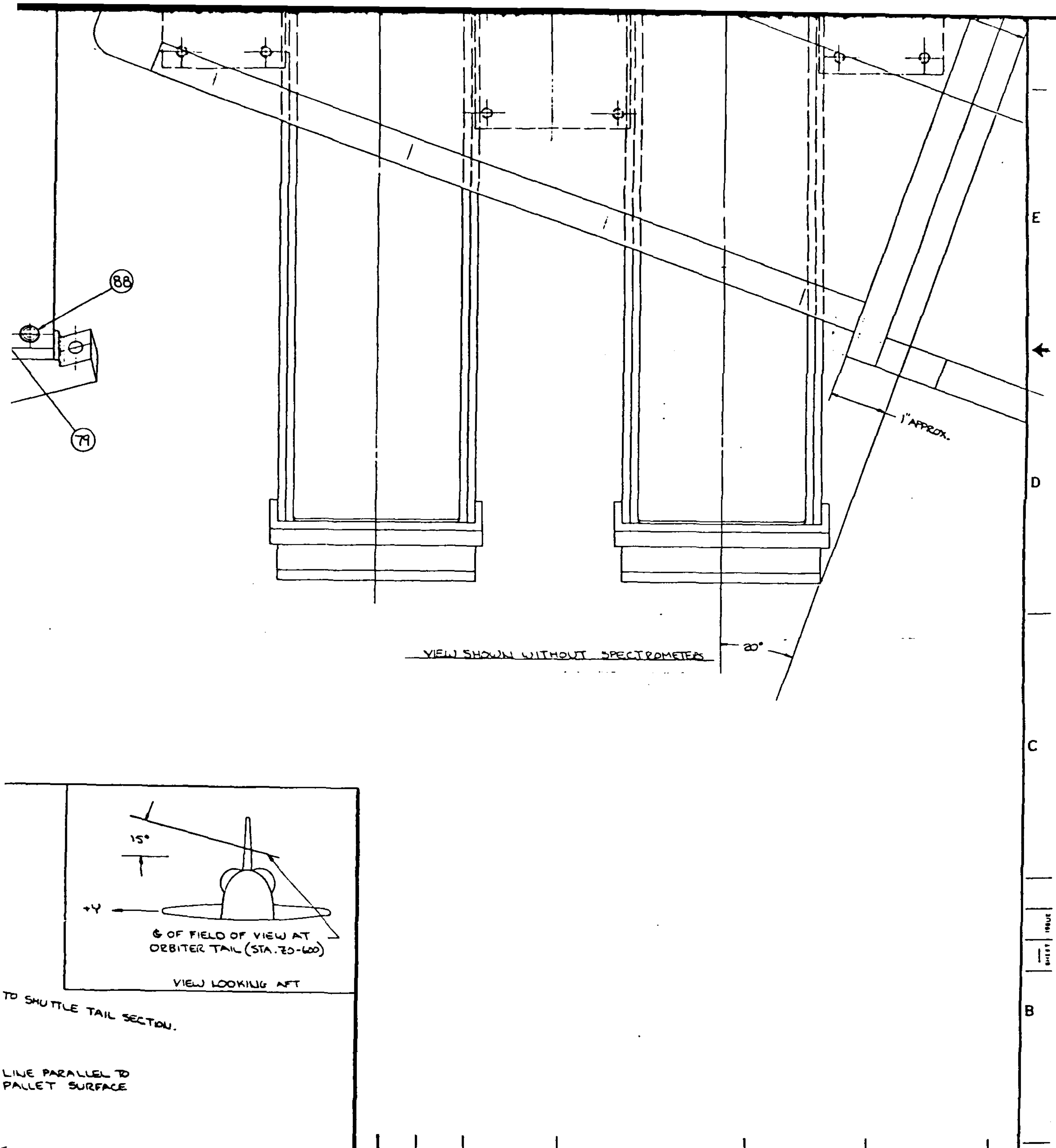
REVISIONS				
SYM	ZONE	DESCRIPTION	DATE	APPROVED
		SEE SHT. 1 FOR ECO LISTING.		





21° MAX. TO ALLOW FOR INSTRUMENT SWING
16½" MAX. TO ALLOW FOR INSTRUMENT SWING.
SPECTROMETER/TELESCOPE - NO. 1
16½
SPECTROMETER/TELESCOPE - NO. 2
ELECTRONICS AREA APPROX. 11½ x 6⅝ x 13"
15½°
MOUNTING PLATE OF SPECTROMETER/TELESCOPE
8
7
6


88
79
15°
VIEW
OPE - NO. 1
PIVOT
75° SCAN ANGLE
20
16½
ONICS AREA
11½ x 6⅝ x 13
15½°
8½
11½"
6½"
16½"
28"
7
15°
20°
TO SHUTTLE TAIL SECTION.
LINE PARALLEL TO PALLET SURFACE
PALLET MOUNTING SURFACE
TING PLATE OF
ROMETER/TELESCOPES
+Y
AFT
MOUNTING CONFIGURATION
15°
+Y
℄ OF FIELD OF VIEW AT ORBITER TAIL (STA. Z0-600)
VIEW LOOKING AFT
QTY. REQ'D.
CODE IDENT
TOLERAN
ENGLISH
INCHES
FRAC. ± 1/64
DECIMALS:
.XXX ± .005
.XXXX ± .0005
ANGLES ±
MATERIAL
FINISH
5
4
3



QTY. REQ'D.	CODE IDENT.	PART NO. OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION	MATERIAL DESCRIPTION	MATERIAL SPECIFICATION	ITEM NO.

LIST OF MATERIAL

TOLERANCES
ENGLISH: INCHES; FRAC. ± 1/64; DECIMALS: .XXX ± .005, .XXXX ± .0005; ANGLES ±
METRIC: MILLIMETERS; DECIMALS: .X ± .1, .XX ± .01

MATERIAL

FINISH

	SIGNATURE	DATE
DRAWN	R. EPPIG	6-1-84
CHECKED		
APPROVED		
APPROVED		
APPROVED		
RELEASED		

RSI RESEARCH SUPPORT INSTRUMENTS INC
COCKEYSVILLE, MARYLAND

DUAL SCANNING SPECTROMETER ASSEMBLY
– AFGL HUP PROGRAM, DR HUFFMAN

CODE IDENT NO	SIZE		ISSUE
56123	E	0-224-0-1	C

SCALE 1:1 SHEET 2 OF 2

Fig. 6

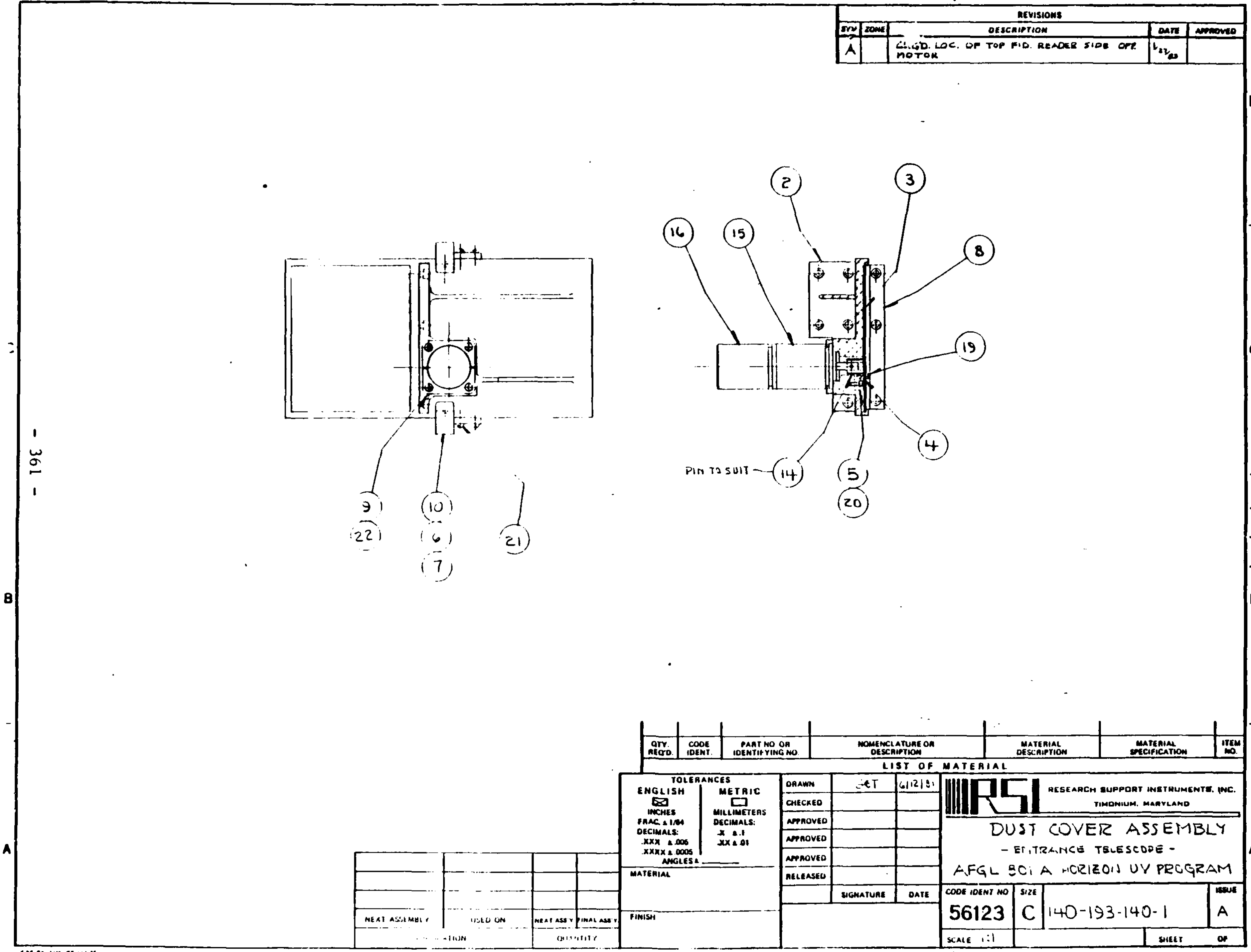
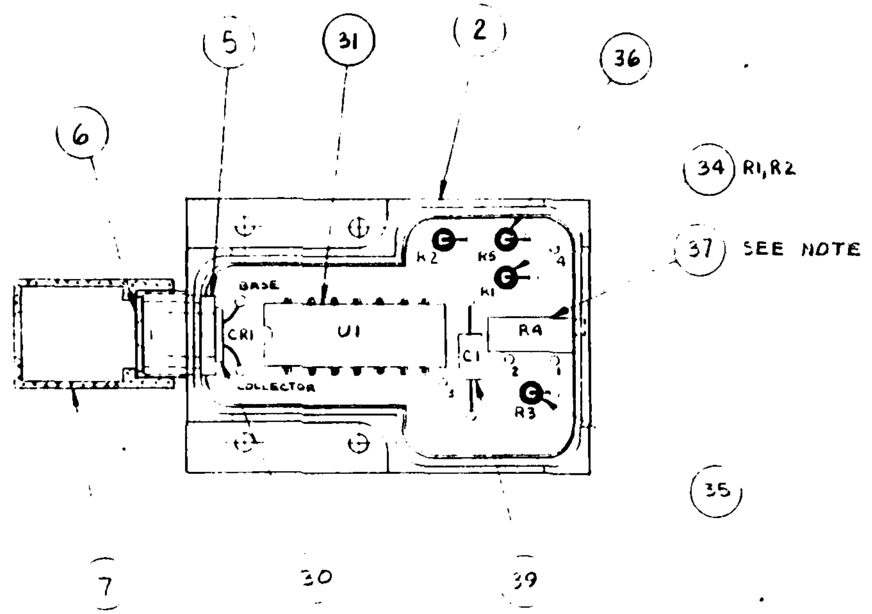

REVISIONS
SYM | ZONE | DESCRIPTION | DATE | APPROVED
A | | CHGD. LOC. OF TOP FID. READER SIDE OPP. MOTOR
PIN TO SUIT
QTY. REQ'D. | CODE IDENT. | PART NO OR IDENTIFYING NO | NOMENCLATURE OR DESCRIPTION | MATERIAL DESCRIPTION | MATERIAL SPECIFICATION | ITEM NO.
LIST OF MATERIAL
TOLERANCES
ENGLISH
INCHES
FRAC. ± 1/64
DECIMALS:
.XXX ± .005
.XXXX ± .0005
ANGLES ±
METRIC
MILLIMETERS
DECIMALS:
.X ± .1
.XX ± .01
MATERIAL
FINISH
DRAWN
CHECKED
APPROVED
APPROVED
APPROVED
RELEASED
SIGNATURE
DATE
NEXT ASSEMBLY
USED ON
NEXT ASSY
FINAL ASSY
QUANTITY
RSI
RESEARCH SUPPORT INSTRUMENTS, INC.
TIMONIUM, MARYLAND
DUST COVER ASSEMBLY
- ENTRANCE TELESCOPE -
AFGL 801 A HORIZON UV PROGRAM
CODE IDENT NO
56123
SIZE
C
140-193-140-1
ISSUE
A
SCALE 1:1
SHEET
OF

Fig. 7

REVISIONS				
SYM	ZONE	DESCRIPTION	DATE	APPROVED



NOTE-

1. POTENTIOMETER MUST BE MOUNTED 3/64 OFF OF BOARD AFTER SOLDERING

QTY REQ'D.	CODE IDENT.	PART NO. OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION	MATERIAL DESCRIPTION	MATERIAL SPECIFICATION	ITEM NO.

LIST OF MATERIAL

TOLERANCES

ENGLISH ☐ INCHES FRAC. ± 1/64 DECIMALS: XXX ± .005 XXXX ± .0005 ANGLES ±

METRIC ☐ MILLIMETERS DECIMALS: .X ± .1 .XX ± .01

MATERIAL

FINISH

	SIGNATURE	DATE
DRAWN	B NICHOLAS	9/18/81
CHECKED		
APPROVED		
APPROVED		
APPROVED		
RELEASED		

RESEARCH SUPPORT INSTRUMENTS, INC.
TIMONIUM, MARYLAND

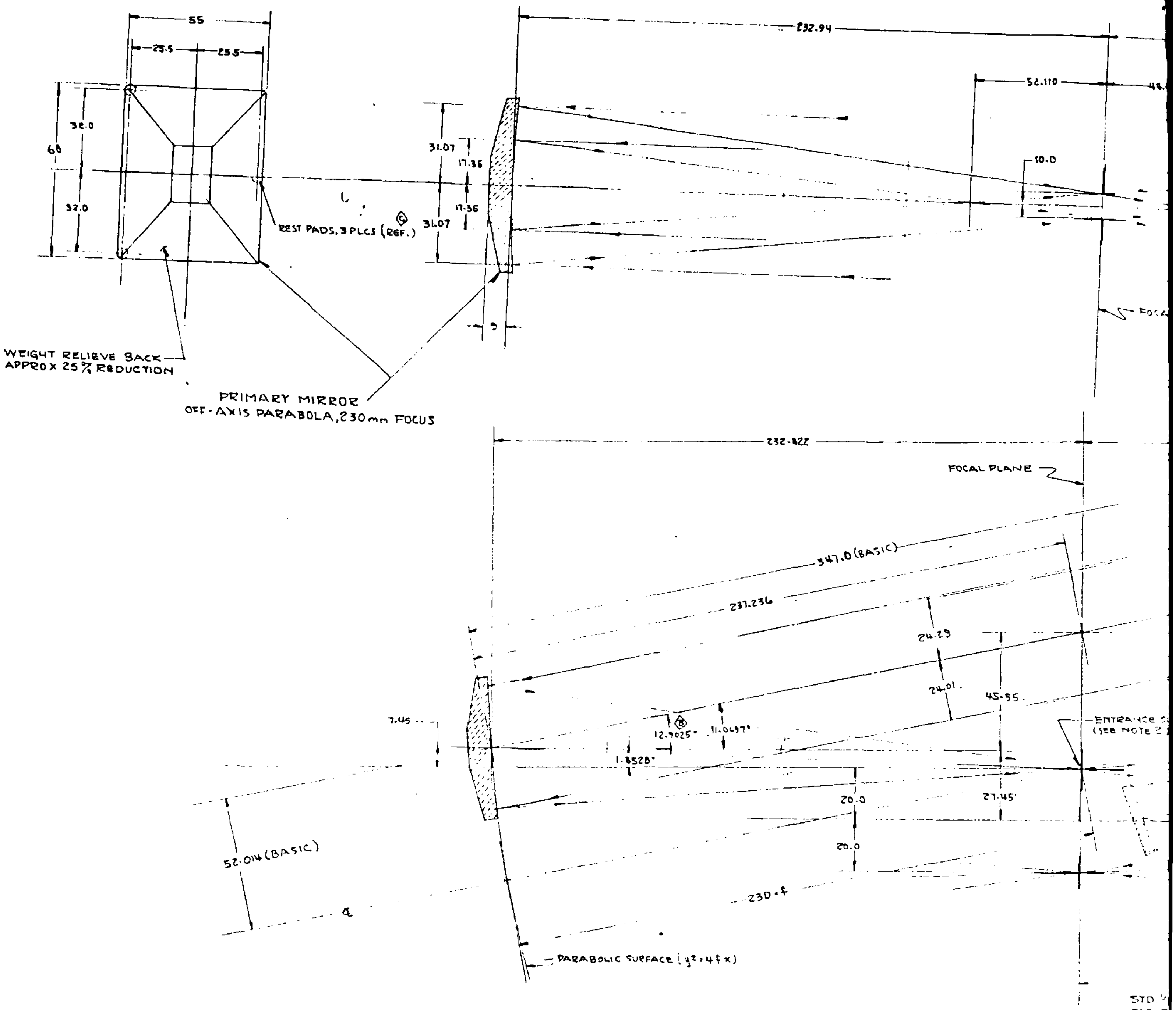
SUN SENSOR ASSEMBLY

AFGL 801A SHUTTLE SETS-PALLET

CODE IDENT NO.	SIZE		ISSUE
56123	B	680-193-0-1	
SCALE :1		SHEET 1 OF 1	

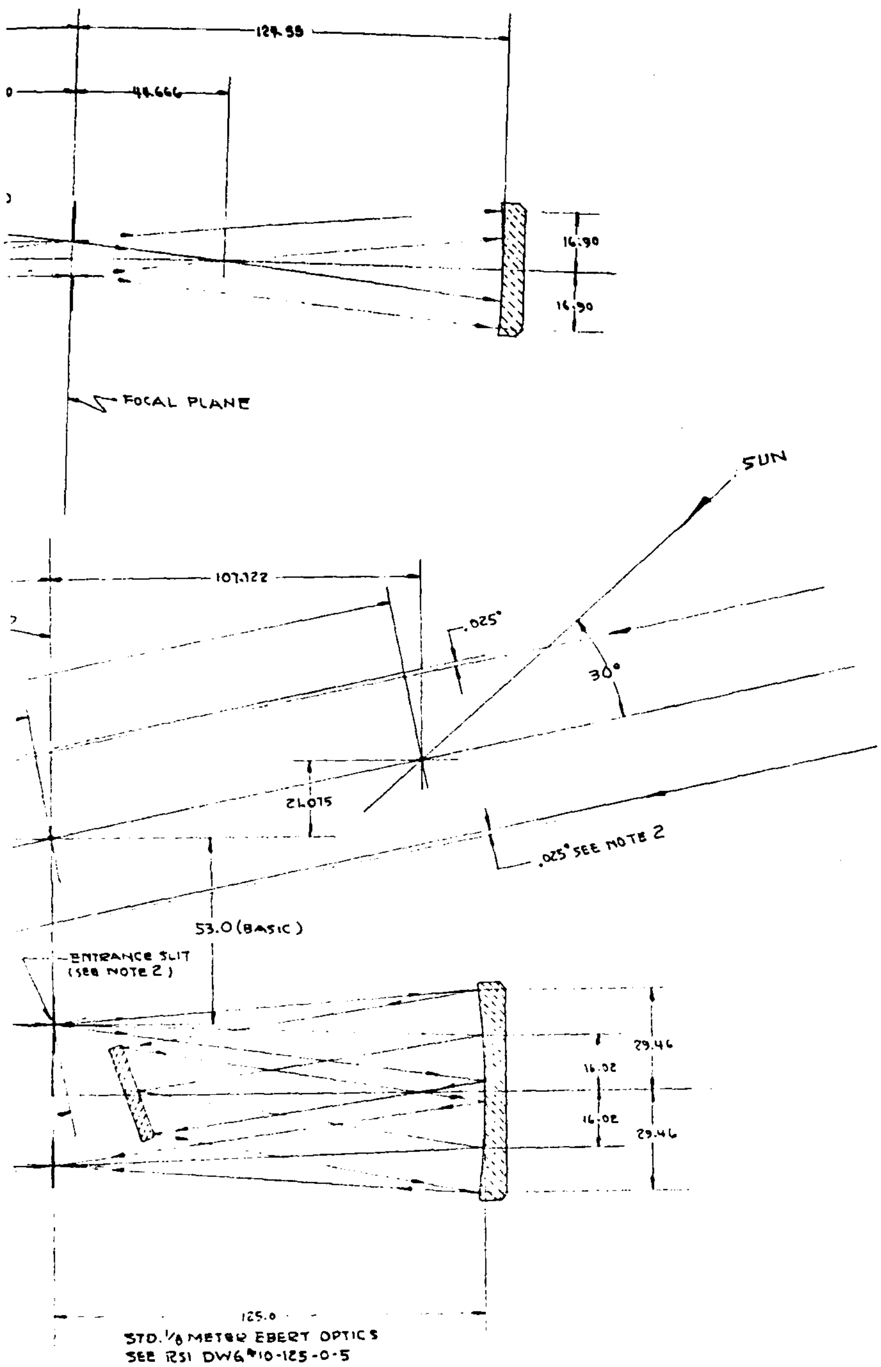
NEXT ASSEMBLY	USED ON	NEXT ASS'Y	FINAL ASS'Y
APPLICATION		QUANTITY	

Fig. 8



NOTES
1) ALL DIMENSIONS IN MILLIMETERS.
ALL ANGLES DECIMAL DEGREES.
2) ENTRANCE SLIT WIDTH MAINTAINED AT 0.20mm
TO CONTROL .05°(.00087 RAD.) MAX FIELD of VIEW

REV.	BY	DESCRIPTION	DATE
A	JST	UPDATED, PRIM. MIRROR WAS SPHERICAL.	[illegible]
B	MEY	REMOVED FLATS FROM MIRROR REAR (OFF-AXIS PARABOLA)	[illegible]
C	JST	12.9025° WAS 12.899°; ASSOCIATED DIMEN'S CORRECTED BASIC DIMEN'S UNCHANGED REST PAD LOC. ADDED	[illegible]



20 mm
of VIEW.

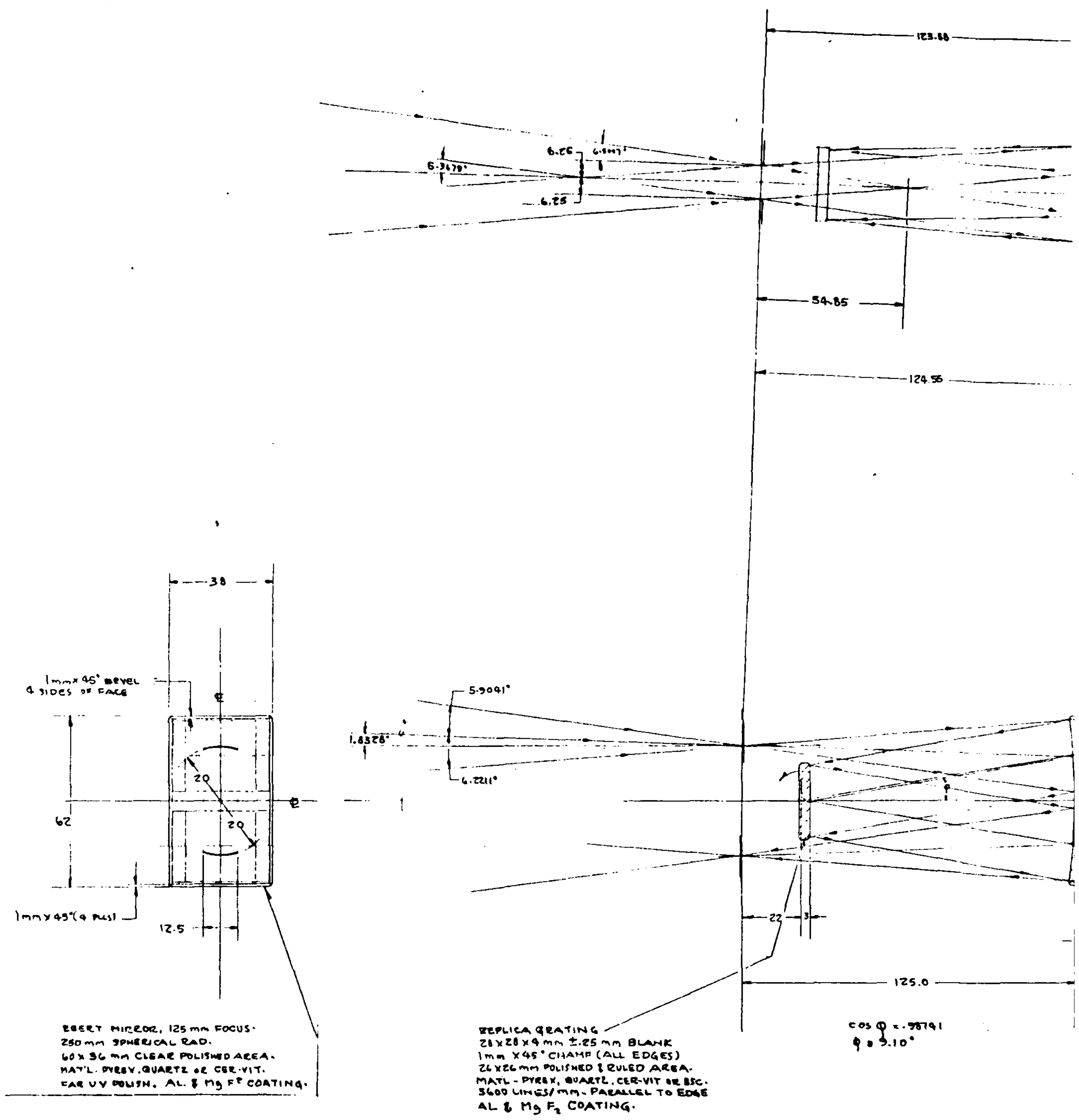
TOLERANCES		DRAWN	J. TRACEY	5/15/7?
ENGLISH INCHES FRAC. ± 1/64 DECIMALS: .XXX ± .005 .XXXX ± .0005 ANGLES ±	METRIC MILLIMETERS DECIMALS: .X ± .1 .XX ± .01	CHECKED		
		APPROVED		
		APPROVED		
		APPROVED		
MATERIAL		RELEASED		
FINISH			SIGNATURE	DATE

RSI RESEARCH SUPPORT INSTRUMENTS, INC. TIMONIUM, MARYLAND

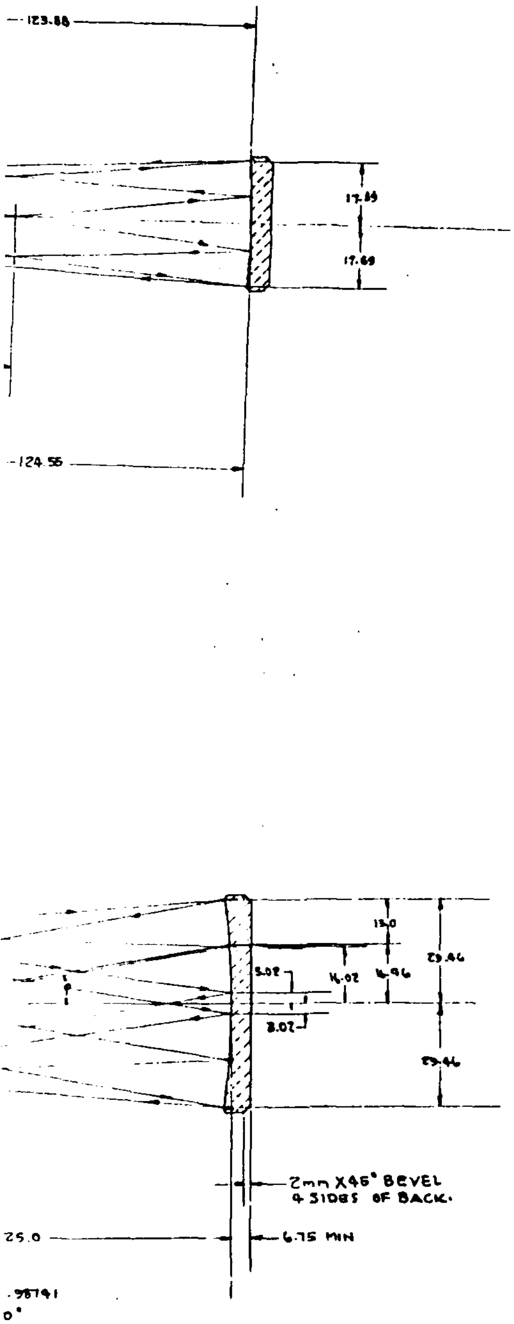
OPTICAL LAYOUT
TELESCOPE & EBERT SPECTROMETER
AFGL S01 HORIZON UV PROGRAM

CODE IDENT NO	SIZE		ISSUE
56123	D	140-193-0-17	C
SCALE FULL SIZE		SHEET 1 OF 1	

Fig. 9


123.88
6.26
6.3479°
6.25
54.85
124.56
38
1mm x 45° BEVEL
4 SIDES OF FACE
20
20
62
1mm x 45° (4 PLS)
12.5
5.9091°
1.8328°
6.2211°
22
3
125.0
EBERT MIRROR, 125 mm FOCUS.
250 mm SPHERICAL RAD.
60 x 36 mm CLEAR POLISHED AREA.
MAT'L. PYREX, QUARTZ OR CER-VIT.
FAR UV POLISH. AL. & Mg F2 COATING.
REPLICA GRATING
28 x 28 x 9 mm ±.25 mm BLANK
1mm X 45° CHAMF (ALL EDGES)
26 x 26 mm POLISHED & RULED AREA.
MATL - PYREX, QUARTZ, CER-VIT OR BSC.
3600 LINES/mm. PARALLEL TO EDGE
AL & Mg F2 COATING.
COS Φ = .98791
Φ = 9.10°

REV.	BY	DESCRIPTION	DATE



TOLERANCES				
ENGLISH: INCHES, FRAC. ± 1/64, DECIMALS: .XXX ± .005, .XXXX ± .0005	METRIC: MILLIMETERS, DECIMALS: .X ± .1, .XX ± .01	DRAWN	J. TRACEY	4/15/75
ANGLES ±		CHECKED		
		APPROVED		
		APPROVED		
		APPROVED		
MATERIAL		RELEASED		
FINISH			SIGNATURE	DATE

RSI RESEARCH SUPPORT INSTRUMENTS, INC.
TIMONIUM, MARYLAND

OPTICAL LAYOUT
.125 m EBERT-FASTIE SPECTROM.

CODE IDENT NO	SIZE		ISSUE
56123	D	10-125-0-5	
SCALE 1:1		SHEET 1 OF 3	

Fig. 10

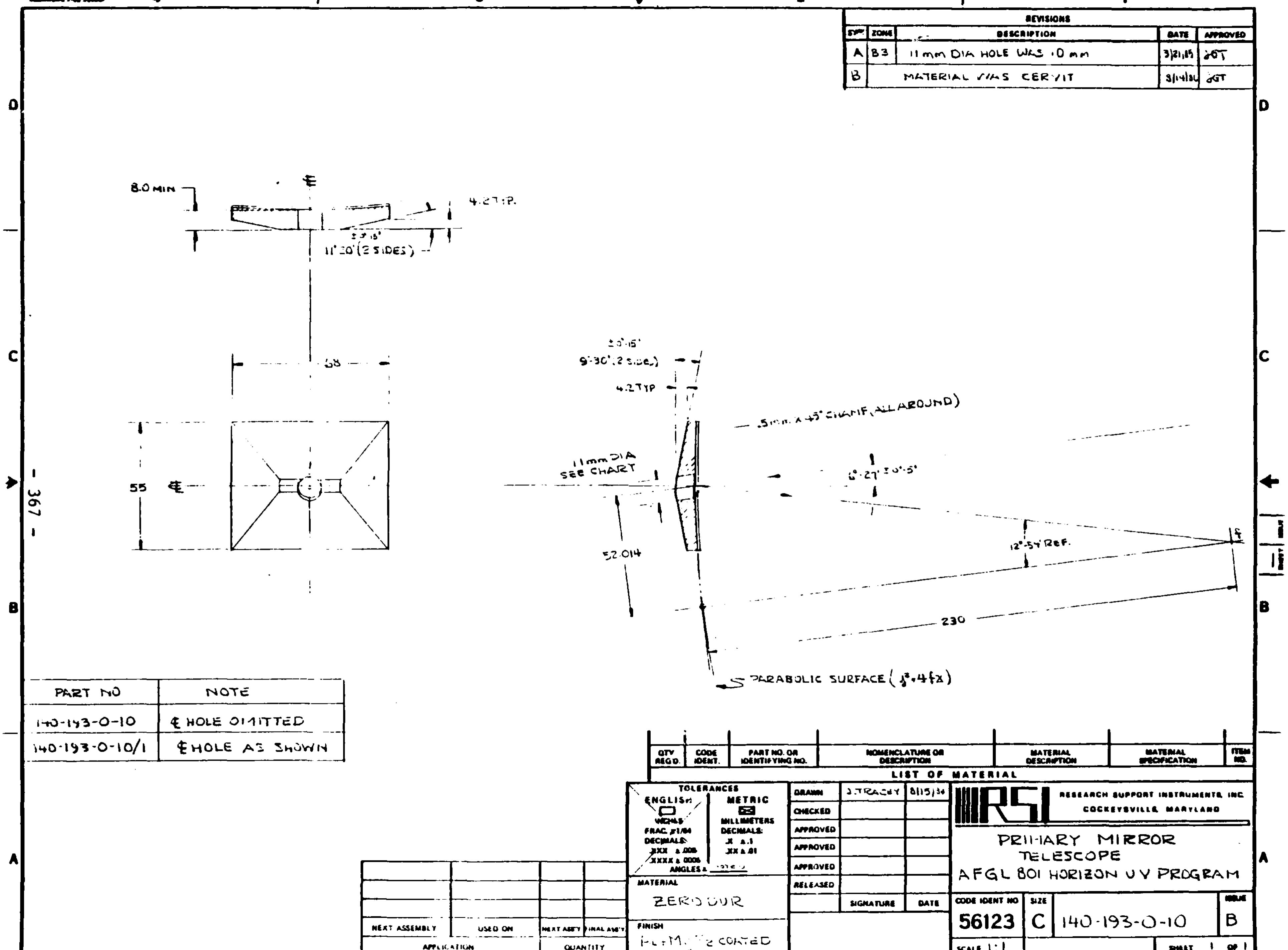

REVISIONS
A B3 11 mm DIA HOLE WAS 10 mm 3/21/85
B MATERIAL WAS CERVIT 3/14/86
8.0 MIN
4.2 TYP.
11°20' (2 SIDES)
68
55
±0°15'
9°30' (2 sides)
4.2 TYP
.5 mm x 45° CHAMF. (ALL AROUND)
11 mm DIA
SEE CHART
6°27' ±0°5'
52.014
12°54' REF.
230
PARABOLIC SURFACE ($y^2=4fx$)
PART NO | NOTE
140-193-0-10 | ℄ HOLE OMITTED
140-193-0-10/1 | ℄ HOLE AS SHOWN
LIST OF MATERIAL
TOLERANCES
ENGLISH
METRIC
MILLIMETERS
MATERIAL
ZERODUR
FINISH
DRAWN J. TRACEY 8/15/84
RESEARCH SUPPORT INSTRUMENTS, INC.
COCKEYSVILLE, MARYLAND
PRIMARY MIRROR
TELESCOPE
AFGL 801 HORIZON UV PROGRAM
CODE IDENT NO 56123
SIZE C
140-193-0-10
ISSUE B
SCALE 1:1
SHEET 1 OF 1

Fig. 11

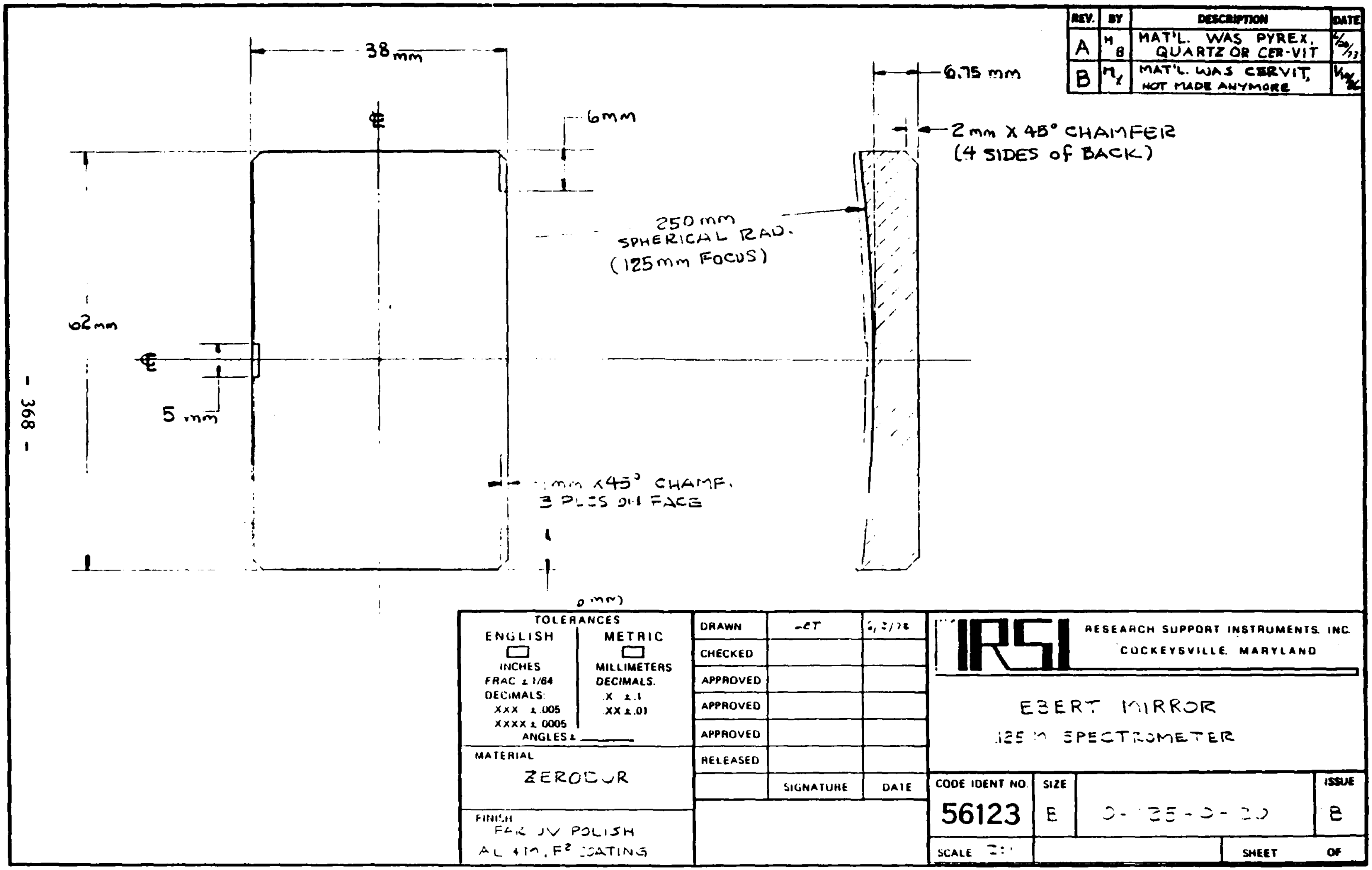

REV. BY DESCRIPTION DATE
A MB MAT'L. WAS PYREX, QUARTZ OR CER-VIT
B MATL. WAS CERVIT, NOT MADE ANYMORE
38 mm
6.75 mm
6mm
2mm X 45° CHAMFER (4 SIDES of BACK)
250 mm SPHERICAL RAD. (125mm FOCUS)
62mm
5 mm
3 PLCS ON FACE
TOLERANCES
ENGLISH
METRIC
INCHES
MILLIMETERS
FRAC ± 1/64
DECIMALS:
XXX ± .005
XXXX ± .0005
DECIMALS.
.X ± .1
.XX ± .01
ANGLES ±
MATERIAL
ZERODUR
FINISH
AL + MgF² COATING
DRAWN
CHECKED
APPROVED
APPROVED
APPROVED
RELEASED
SIGNATURE
DATE
RESEARCH SUPPORT INSTRUMENTS, INC.
COCKEYSVILLE, MARYLAND
EBERT MIRROR
SPECTROMETER
CODE IDENT NO.
56123
SIZE
E
ISSUE
B
SCALE
SHEET
OF

Fig. 12

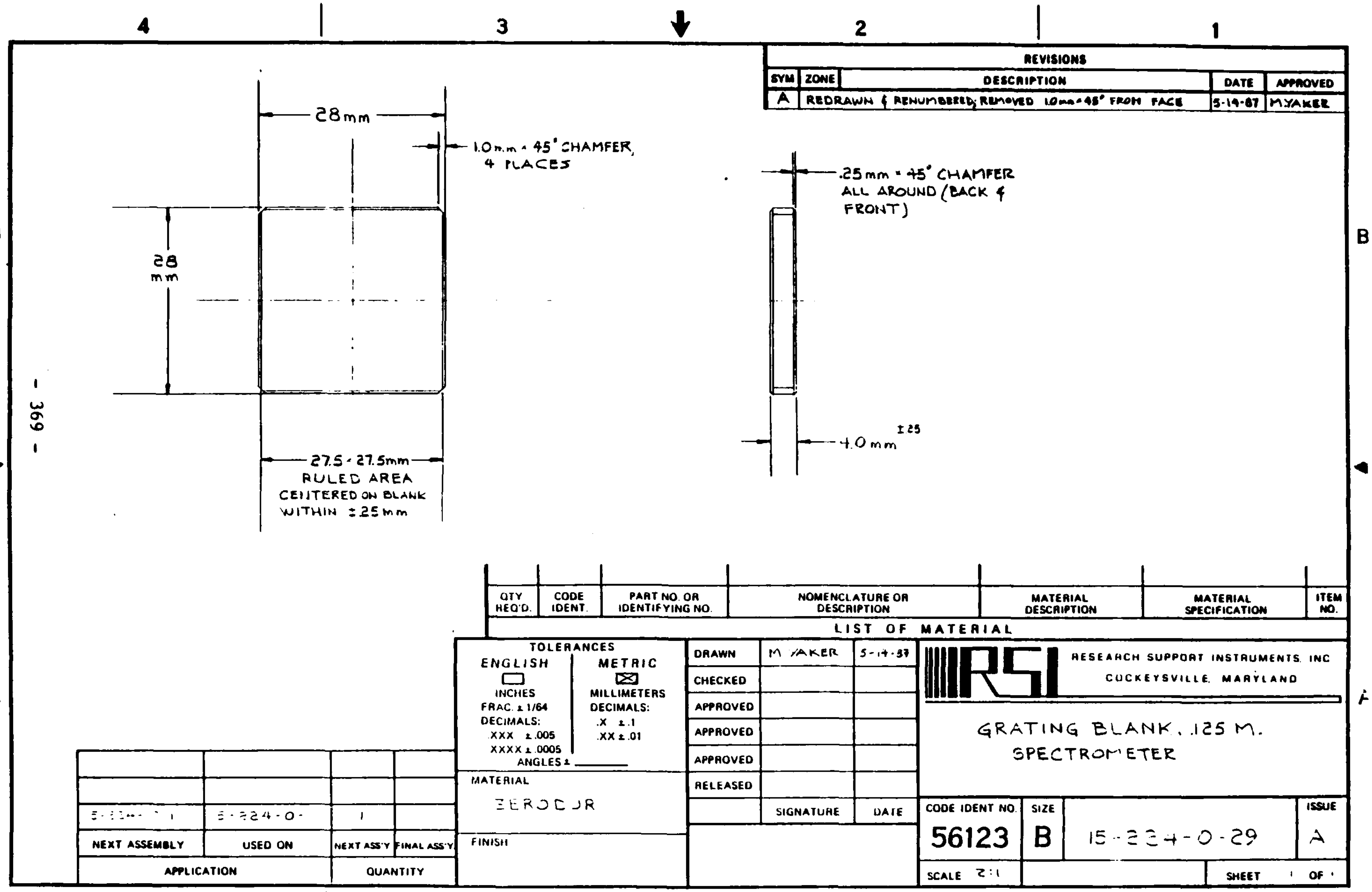
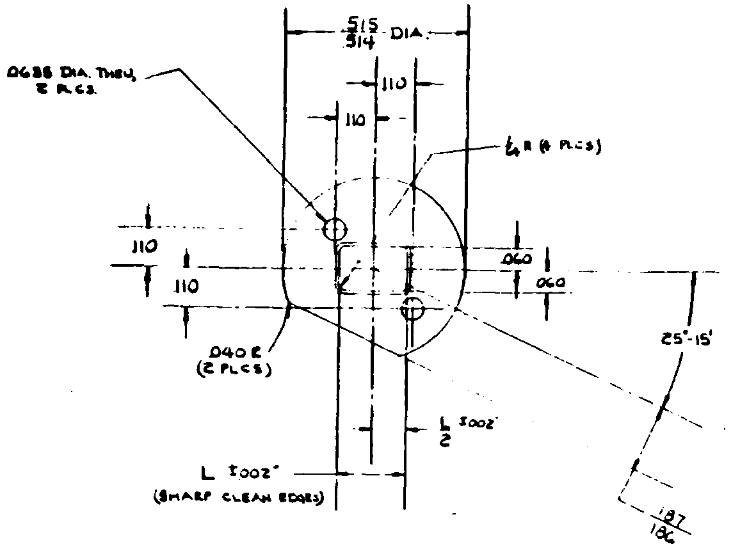
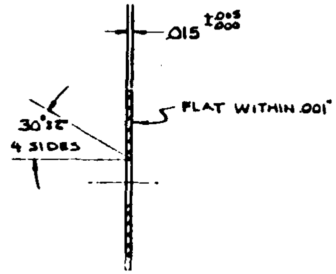

REVISIONS
SYM | ZONE | DESCRIPTION | DATE | APPROVED
A | REDRAWN & RENUMBERED; REMOVED 1.0mm×45° FROM FACE | 5-14-87 | M.YAKER
28mm
1.0mm × 45° CHAMFER, 4 PLACES
28 mm
27.5 × 27.5mm RULED AREA CENTERED ON BLANK WITHIN ±.25mm
.25mm × 45° CHAMFER ALL AROUND (BACK & FRONT)
4.0mm ±.25
QTY REQ'D. | CODE IDENT. | PART NO. OR IDENTIFYING NO. | NOMENCLATURE OR DESCRIPTION | MATERIAL DESCRIPTION | MATERIAL SPECIFICATION | ITEM NO.
LIST OF MATERIAL
TOLERANCES
ENGLISH
INCHES
FRAC. ± 1/64
DECIMALS:
.XXX ± .005
.XXXX ± .0005
ANGLES ±
METRIC
MILLIMETERS
DECIMALS:
.X ± .1
.XX ± .01
MATERIAL
ZERODUR
FINISH
DRAWN M. YAKER 5-14-87
CHECKED
APPROVED
APPROVED
APPROVED
RELEASED
SIGNATURE
DATE
RSI
RESEARCH SUPPORT INSTRUMENTS, INC
COCKEYSVILLE, MARYLAND
GRATING BLANK, .125 M. SPECTROMETER
CODE IDENT NO. 56123
SIZE B
15-234-0-29
ISSUE A
SCALE 2:1
SHEET 1 OF 1
NEXT ASSEMBLY
USED ON
NEXT ASS'Y
FINAL ASS'Y
APPLICATION
QUANTITY
1
4
3
2
1
B
A

Fig. 13

REV	BY	DESCRIPTION	DATE
A	ME Y	REMOVED 17. 7 (180mm) DET. O/M TUBE ONLY 6mm	





PART NO.	SLIT HGT. 'L' MM	INCHES	REMARKS
17-125-0-22/1	2.0	.078	
/2	3¼	.125	
/3	5.0	.197	
/4	1.0	.039	
/5	.10	.004	100μ
/6	6.0	.236	
/7	.5	.0197	500μ
/8	.4	.0157	400μ
/9	.3	.0118	300μ
17-125-0-22/10	.2	.0079	200μ
/11	2.5	.0984	250μ
/12	6.16	.2425	6160μ
/13	.80	.0315	800μ
/14	1.50	.0591	1500μ
17-125-0-22/15	7.0	.2756	700μ
/16	9.0	.3543	900μ

NOTES:

1) LIGHT BEAD BLAST TO DULL MATTE FINISH & BLACK ANODIZE. DO NOT DAMAGE EDGES AT 'L' DIM.

TOLERANCES				
ENGLISH ☒ INCHES FRAC ±1/64 DECIMALS .XXX ±.005 .XXXX ±.0005 ANGLES ± .5°	METRIC ☐ MILLIMETERS DECIMALS .X ±.1 .XX ±.01	DRAWN	M E YAKER	6/4/--
		CHECKED		
		APPROVED		
		APPROVED		
		APPROVED		
MATERIAL 2024-T3 ALUM ALLOY PUNCHED BLANK		RELEASED		
FINISH SEE NOTE 1			SIGNATURE	DATE

IRSI RESEARCH SUPPORT INSTRUMENTS INC COCKEYSVILLE MARYLAND

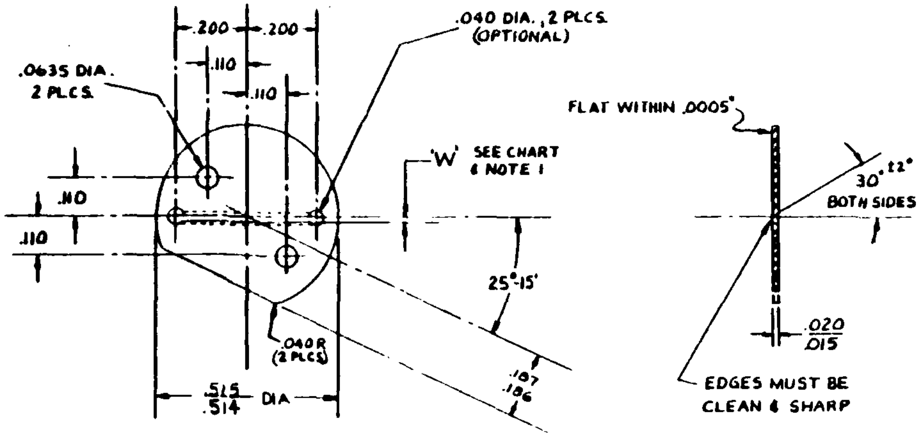
MASK SLIT HEIGHT

CODE IDENT NO	SIZE		ISSUE
56123	C	17-125-0-22	A
SCALE 4/1		SHEET 1 OF 1	

H-E

Fig. 14

REVISIONS				
SYM	ZONE	DESCRIPTION	DATE	APPROVED
A		REDRAWN TO "C" SIZE & ADDED PART NOS. TO CHART	4/23/81	



PART NO.	SLIT 'W' M M	SLIT 'W' INCHES	REMARKS
17-125-0-54/1	.33	.013	333μ
17-125-0-54/2	.5	.0197	500μ
17-125-0-54/3	1.0	.0394	1000μ
17-125-0-54/4	.15	.0060	150μ
17-125-0-54/5	.75	.0295	750μ
17-125-0-54/6	.20	.0079	200μ
17-125-0-54/7	.25	.0098	250μ
17-125-0-54/8	.085	.0033	85μ
17-125-0-54/9	.05	.0020	50μ
17-125-0-54/10	.10	.0039	100μ
17-125-0-54/11	1.50	.0590	1500μ
17-125-0-54/12	.80	.0315	800μ
17-125-0-54/13	1.10	.0433	
17-125-0-54/14	.36	.0141	
17-125-0-54/15	.24	.0095	

NOTES

1)-TOLERANCES FOR SLIT WIDTH- HOLD TO 10% OF 'W' FOR OPENINGS LESS THAN .5MM AND 5% OF 'W' FOR OPENINGS .5MM AND GREATER.

2)-LIGHT BEAD BLAST TO DULL MATTE FINISH BEFORE SIZING 'W'.

QTY. REQ'D.	CODE IDENT.	PART NO OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION	MATERIAL DESCRIPTION	MATERIAL SPECIFICATION	ITEM NO.

LIST OF MATERIAL

TOLERANCES
ENGLISH: INCHES; FRAC. ± 1/64; DECIMALS: .XXX ± .005, .XXXX ± .0005
METRIC: MILLIMETERS; DECIMALS: .X ± .1, .XX ± .01
ANGLES ± ±/5

MATERIAL: ALUM. 6061-T6 PUNCHED BLANK

FINISH: SEE NOTE 2

	SIGNATURE	DATE
DRAWN	B. NICHOLAS	4/23/81
CHECKED		
APPROVED		
APPROVED		
APPROVED		
RELEASED		

NEXT ASSEMBLY	USED ON	NEXT ASS'Y	FINAL ASS'Y
APPLICATION		QUANTITY	

RSI RESEARCH SUPPORT INSTRUMENTS, INC. TIMONIUM, MARYLAND

SLIT WIDTH (STR'GHT)
.125 M. SPECTROMETER

CODE IDENT NO	SIZE		ISSUE
56123	C	17-125-0-54	A
SCALE 4:1		SHEET 1 OF 1	

Fig. 15